AD-A146 681 THE APPLICATION OF ARTIFICIAL INTELLIGENCE TO CONTRACT MANAGEMENT(U) CONSTRUCTION ENGINEERING RESEARCH LAB (ARMY) CHAMPAIGN IL T A KRUPPENBACHER AUG 84 1/6

UNCLASSIFIED CERL-TM-P-166 F/G 9/1 NL

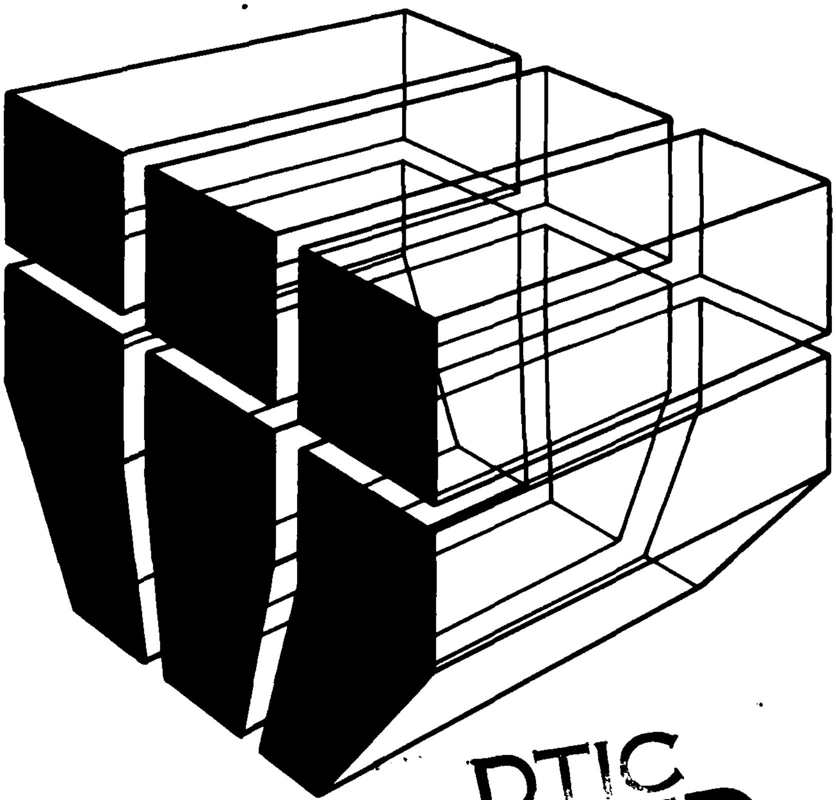
US Army Corps
of Engineers
Construction Engineering
Research Laboratory

TECHNICAL MANUSCRIPT P-166
August 1984

(12)

THE APPLICATION OF ARTIFICIAL INTELLIGENCE TO CONTRACT MANAGEMENT

by
Timothy A. Kruppenbacher



Approved for public release; distribution unlimited.

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER CERL-TM-P-166	2. GOVT ACCESSION NO. AD-A146681	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) THE APPLICATION OF ARTIFICIAL INTELLIGENCE TO CONTRACT MANAGEMENT		5. TYPE OF REPORT & PERIOD COVERED Final
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Timothy A. Kruppenbacher		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS U.S. ARMY CONSTRUCTION ENGINEERING RESEARCH LABORATORY P.O. BOX 4005, CHAMPAIGN, IL 61820		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS		12. REPORT DATE August 1984
		13. NUMBER OF PAGES 523
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

Copies are available from the National Technical Information Service
Springfield, VA 22161

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

artificial intelligence
contract administration
construction
claims

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

The field of artificial intelligence (AI) is rapidly developing new techniques to improve the environment in which systems capable of simulating intelligent behavior can be developed. As these new developments occur, additional areas for possible application of these new techniques become apparent.

The area of contract management currently holds many opportunities for the development of expert systems which are capable of assuming the role of a legal consultant on matters pertaining to claim analysis. To demonstrate the

DD FORM 1473 1 JAN 73 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

BLOCK 20. (Cont'd)

feasibility of this type of expert system, the Differing Site Condition Analysis System (DSCAS) has been developed. DSCAS, built within the ROSIE programming environment, is capable of performing the legal analysis of a differing site condition (DSC) claim. The DSCAS program is based on logic which is patterned after the decision process used by a lawyer to analyze the DSC claim. DSCAS provides a very user-friendly environment in which the analysis is performed and a number of desirable features. Two of the most desirable features are: its ability to make assumptions and continue the analysis if an answer is unknown and its ability to explain the reason behind concluding that the contractor will not be allowed entitlement.

DSCAS has been tested on a number of cases to determine its abilities and limitations. The tests revealed that DSCAS does indeed possess the ability to accurately analyze a claim. However, much work is still necessary to refine the logic in order to allow DSCAS to ask questions throughout the analysis which require answers based on factual knowledge rather than on legal knowledge. Based on the results of testing the current version, DSCAS demonstrates that the techniques of AI can be successfully applied to contract management.

FOREWORD

This project was funded by the U.S. Army Construction Engineering Research Laboratory (USA-CERL) under the Graduate Research Assistantship Program. This manuscript was presented to the faculty of the graduate school of the University of Colorado in partial fulfillment of the requirements for the degree of Master of Science, Department of Civil, Environmental, and Architectural Engineering, 1984.

Appreciation is expressed to Associate Professor James Diekmann for his advice, interest, and effort, and to the typists (especially Cindy Musselmann) for their effort and time.

Mr. Edward A. Lotz is Chief of USA-CERL-FS. COL Paul J. Theuer is Commander and Director of USA-CERL, and Dr. L. R. Shaffer is Technical Director.

CONTENTS

IV. SEARCH

V. PROBLEM SOLVING

TABLES

Table

FIGURES

Figure

CHAPTER I

INTRODUCTION

The purpose of this thesis is to demonstrate the feasibility of applying the techniques of artificial intelligence (AI) to contract management. The feasibility of this application is demonstrated through the development of an expert system which is capable of modelling the thought processes implemented by a lawyer during the analysis of the differing site condition claim. Before explaining the work of this thesis a brief description of two of the fields being brought together in this thesis is essential.

The aspect of contract management implemented by this thesis is that of construction claims analysis. The differing site condition claim has been selected as the problem domain for this application of AI techniques. The differing site condition claim is a contractually granted remedy for additional expenses incurred by the contractor due to the occurrence of a differing site condition during the performance of a construction contract. A differing site condition occurs when the contractor encounters a latent physical condition at the site which is materially different from those conditions indicated by the contract or materially different from those conditions which could ordinarily be expected.

AI is a relatively young field of science concerned with the design and construction of computer systems which exhibit the characteristics of intelligent behavior. A number of AI systems have been developed which are capable of performing specific tasks requiring intelligent behavior. Some fields to which the techniques of AI have been applied include robotics, pattern recognition, medical diagnosis and geologic exploration. AI systems, such as those developed for medical diagnosis and geologic exploration, act as intelligent agents or consultants and are referred to as expert systems.

1.1 Overview of the Work

The AI system developed as part of this thesis is a prototype of an expert system for the analysis of a differing site condition (DSC) claim. The Differing Site Condition Analysis System (DSCAS) is based on a model of the decision process used by a lawyer for the analysis of the DSC claim. The decision process implemented by DSCAS is based on twenty-two logic modules, each pertaining to a specific concept essential to the analysis of the claim. The DSCAS program implements an intricate system of controls to analyze the DSC claim by gathering all the information pertinent to the claim known by the user. DSCAS then analyzes the information which has been gathered drawing the appropriate conclusions and requesting additional information as needed. DSCAS attempts to analyze at any one time only those facts or answers given by the user which pertain to a single concept or logic module. Once

all the conclusions have been drawn the next appropriate logic module is invoked. The order in which logic modules are invoked is determined by the conclusions which have been drawn from previous logic modules. DSCAS also allows the user to perform the analysis contained by a single logic module, if desired.

The intent of a tool such as DSCAS is to provide field personnel on a construction jobsite with the expertise of a legal consultant without actually having a legal consultant present. If, for example, a contractor asserts that a differing site condition has been encountered the contracting officer (CO) need only access the DSCAS program for the legal assistance necessary to properly handle the analysis of the claim. During the session DSCAS leads the CO through the steps essential to the analysis of the claim by asking questions which gather information pertinent to the claim. As the session progresses DSCAS records all the information given by the CO and draws any appropriate conclusions. If the CO is unable to answer any of the questions asked, DSCAS makes an assumption and continues with the analysis. Once the CO has entered all the information he is currently aware of he is able to end the analysis session.

At the end of the analysis session the CO will receive a copy of all facts which are known as well as a list of all unknowns and the corresponding assumptions which were made. The list of unknowns can then be used to determine what additional information is necessary to complete the analysis of the claim. After seeking out this additional information the CO can hold another session with

based in an attempt to provide the reader with sufficient understanding of previous work in these fields. Chapters two through eight concentrate on the important concepts essential for understanding each of the fields brought together in this thesis, i.e., AI, contract management and computerized legal analysis systems. Since most readers are generally unfamiliar with the concepts of AI, chapters two through six compose a primer for AI by reviewing and explaining some of the basic concepts essential to the AI systems which have been developed. Chapter seven presents the aspect of contract management to which the techniques of AI are being applied, i.e., the differing site condition claim, and discusses some of the important considerations which are necessary when analyzing the DSC claim. Chapter eight then reviews a number of computerized legal analysis systems which have been developed and suggests characteristics which the ideal legal analysis system might exhibit.

Chapter nine presents the ROSIE environment in which the DSCAS program has been constructed. ROSIE is the Rule Oriented System for Implementing Expertise which was developed by the Rand Corporation as a programming environment for the construction of expert systems. The discussion of ROSIE in chapter nine explains the features and capabilities provided bv the ROSIE environment as well as some of its limitations.

Chapter ten deals specifically with the DSCAS program explaining the logic on which it is based and the structure of the program itself. The first section discusses the development of the

logic on which the analysis performed by DSCAS is based. This section also explains the advantages and disadvantages of DSCAS's method of analysis as well as the rationale behind the current structure of the logic. The second section discusses both the development of the program and the rationale behind the current configuration of the system. The DSCAS program is explained by examining in detail each of the six major components and the interrelationships between these components.

The final chapter, chapter eleven, deals with the test results explaining how the tests were run as well as the problems, limitations and accomplishments observed during the testing of DSCAS. In addition, this chapter draws a number of conclusions based on the work of this thesis and presents some recommendations for further development of the DSCAS program.

CHAPTER II

ARTIFICIAL INTELLIGENCE

2.1 Definition of Artificial Intelligence

Artificial Intelligence (AI) is the science of the design and construction of computer systems which exhibit the characteristics of human behavior. By examining the literal definition of Artificial Intelligence one is able to discern the emphasis of the current work in this young field. Webster's New Collegiate Dictionary (1977) defines the word "artificial" to mean "humanly contrived . . . brought into being not by nature but by human art or effort."[1] Intelligence, however, is not so easily defined since there exists no precise criteria by which it might be determined. For our purpose we will accept Webster's (1977) definition which refers to intelligence as "the ability to learn or understand or to deal with new and trying situations, the skilled use of reason . . . the ability to apply knowledge."[2] The research of AI, therefore, deals primarily with the development of man-made systems, which exhibit attributes of intelligent human behavior, such as the ability to learn, to understand, to use reason and to apply knowledge to various situations to achieve a solution.

---

[1] Webster's New Collegiate Dictionary, 1977 ed., s.v. "artificial."

[2] Ibid., s.v. "intelligence."

2.2 History of Artificial Intelligence

AI is a very young field of science having its beginnings only about thirty years ago. The current ideas for modelling intelligent human behavior developed from the fields of engineering, psycholinguistics, computer science and cognitive psychology (Lenat 1978). The initial concepts came from a combination of ideas dealing with mathematical logic and computation. These concepts were built upon by contributors such as Frege, Whithead, Russel and Tarski, who developed concepts for the use of simple frameworks to represent forms of reasoning. The work of Weiner, McCulloch and others contributed principles of cybernetics and self-organizing systems. Others contributed ideas dealing with representations of the function of the nervous system through information theory and control theory. It was not until the creation of the "intelligent" machines, which could actually begin to carry out and test these theories, that AI had its actual beginning. The pioneers in the development of the initial computing and non-computing machines include Turing, Babbage and vonNeumann (Barr and Feigenbaum 1981).

Alan Turing is said to be the "Father of AI." He developed what is commonly referred to as the "Turing Test" or, as he called it, the "Imitation Game." This was the first non-numerical model of computation. Turing was one of the first to argue strongly for machine intelligence and developed his model as an answer to the question "Can machines think?" (Lenat 1978). Since the time of Turing, AI has come to be regarded as a separate field of computer

science and psychology. Researchers of AI have developed computer programs which have demonstrated the ability to perform as well as humans in certain instances (Waltz 1982). The ultimate goal of scientists working in the field of AI is to develop expert consultants or intelligent agents which will one day be self-sufficient machines (Bernhard 1980). Additionally, behavioral scientists hold an interest in AI research, hoping to attain new insights for the development of detailed models of the brain from the use of many AI processes which model components of human thought.

2.3 Approaches to Artificial Intelligence

Researchers have taken three approaches to their study of AI. They include artificial networks, artificial evolution and heuristic programming (Slagle 1971). Each approach is based on a different view of the process or mechanics of human intelligence. Artificial networks involve the representation of concepts as a large number of simple elements and their interconnections. This approach is based on the theory that natural thought processes occur as a result of the interaction of neural networks. By adding to the network and adjusting the interconnections within the network this approach has the ability to "learn" and adapt from experience. A marked disadvantage of the network approach is that it is somewhat impractical due to the sheer magnitude of elements necessary to represent the human brain.

Artificial evolution is an approach which attempts to develop systems through mutation and selection, or the process of evolution.

This approach is based on the view that human intelligence has developed through evolution, i.e., natural intelligence comes about from the formation of an initial concept, then subsequent mutations are performed and natural selection occurs resulting in the refined concept. At present this approach has only been able to evolve enough to solve extremely simple problems. The two major problems which have been encountered are: (1) the process of natural evolution is not completely understood, and (2) the system must be able to undergo the evolutionary process much more rapidly than natural evolution if any significant progress is to be made.

Heuristic programming is quite possibly the most popular approach. In current research much success has been achieved using this approach which involves the use of heuristics as guides for the processes which solve the problem at hand. By way of definition a heuristic is a "rule of thumb," strategy, pet theory or general inference procedure which when used in combination with factual knowledge serves to improve the efficiency of a system and gives that system the ability to exhibit intelligent human behavior. This approach is based on the concept that human thought processes are a procedure of search guided by heuristics, which in turn discover other heuristics to guide the on-going search process within our minds. Heuristic programs have been developed which can "learn" from their experiences and some have the ability to solve a variety of different types of problems. There are two types of heuristics: those which are specific, limited to one domain, and those which are general which can function in a number of domains. However, one disadvantage to using heuristic

programming does exist. Heuristics do not guarantee that the optimal result will be found.

2.4 What Lies Ahead?

The following chapters, chapters three through six, will discuss the basic elements and techniques used in many AI systems. These chapters also discuss some of the current systems developed and their corresponding subfields of AI research. The base elements and techniques essential to most AI programs which will be included are:

1. Knowledge Representation—the methods of representing knowledge in order that it can be used both efficiently and easily by the system and the user.
2. Search—the methods used to explore the information represented within the system according to some predetermined technique.
3. Problem Solving—the application of various methods of deduction, induction and recognition to achieve a solution to the problem at hand.

Additionally, as part of the discussion of these basic techniques some of the important advantages and disadvantages which have become apparent from their use will be explored.

The subfields of AI research which will be briefly discussed include: game playing and problem solving, theorem proving, language processing and understanding, learning and planning, programming languages and expert systems. Although many of the basic concepts

involved in each of these subfields will be explored, the emphasis will be placed on discussing the current systems which have been developed that exemplify the work accomplished in these subfields. It is important to note, however, that the systems mentioned may have been based on a number of AI concepts, not just those of the subfield under which that system is discussed.

The treatise of AI is meant to establish the fundamentals of current techniques and to determine the "state of the art" of systems presently in use. As a basis for comparison, advantages, disadvantages and some of the requirements for the application of each technique are identified. Finally, based on the capabilities of the techniques and the dimensions of our problem type the thesis develops a working model of an AI system.

CHAPTER III

KNOWLEDGE REPRESENTATION

3.1 Definition and Overview

Knowledge representation is an essential technique used in artificial intelligence systems to represent large amounts of data by the most effective method. Knowledge representation can be regarded as a scheme used to construct an optimal arrangement of knowledge to facilitate the manipulation and interpretation of that knowledge by the system resulting in intelligent behavior. There exists a host of possible representation schemes; however, most of them can be classified within three major divisions (Mylopoulos 1980).

1) Declarative representation schemes place emphasis on describing the "world" which is being dealt with in terms of objects and events, i.e., facts and the relationships existing between these facts. Declarative schemes can be subdivided into logical schemes and network schemes, such as semantic networks, for representing knowledge.

2) Procedural representation schemes also represent the "world" as objects and events. However, in so doing they are represented as procedures which give the system directions as to how these facts are to be used. Procedural schemes include production systems and

methods which use guided inference, such as pattern-directed inference systems. Although no representation scheme fits completely into one of the two schemes already mentioned, most systems are regarded as being founded on the principles of one scheme or the other.

3) The third classification of representation schemes results from attempts to combine the best features of both the declarative and procedural representation schemes. Within this classification which will be termed "combined schemes," the most important systems for representation are frames and scripts. Frame schemes are structured so that each frame contains a description of a situation, i.e., objects and events, and information regarding the use of the frame (Winston 1977). Frames which are linked together will give a schematic description of an event. Scripts are quite similar to frames in that they are structured in the same manner yet they are ideally suited for representing sequences of events (Barr and Feigenbaum 1981).

3.2 Knowledge and Its Uses

Before elaborating on the above representation schemes, the nature of knowledge to be represented and the uses of the knowledge represented will be discussed. Knowledge, relating to AI work, exists in four basic types: knowledge of objects, knowledge of events, knowledge of performance and knowledge of self (Barr and Feigenbaum 1981). Knowledge of self is more commonly referred to as meta-knowledge, which is the knowledge a system contains about itself, i.e., a system's awareness of what it knows. The ability

of AI systems to treat these types of knowledge is essential for intelligent behavior.

The knowledge representation scheme of a system must not only be capable of representing these various types of knowledge but must also be able to represent this knowledge in a manner which assists in acquisition, retrieval and reasoning processes (Barr and Feigenbaum, 1981). Acquisition of knowledge deals not only with gaining new knowledge through some natural or convenient means but also includes the classification of this new knowledge to facilitate positioning it within the existing knowledge base. During this phase of acquisition the new knowledge must also interact with the existing knowledge to improve the overall knowledge of the system.

Once new knowledge has become established within the knowledge base, the system or the user must be able to retrieve this information. Retrieval is usually accomplished by "lumping" for data which is frequently used together or by "linking" for the use of data which requires the use of another piece or series of related pieces of knowledge. Lumping is an internal relationship whereby knowledge is grouped together within the knowledge base. Linking, on the other hand, gives one bit of knowledge the ability, when acted upon, to call up additional bits of knowledge which in turn call up other bits of knowledge in a series fashion. The effect of lumping and linking is to reduce the effort and search time required. This, in turn, increases the amount of work which can be accomplished by the system in a given unit of time.

For a system to exhibit intelligent behavior it must have the ability to reason with the knowledge it contains. Reasoning can be viewed as the system's ability to piece bits of knowledge together to form new bits of knowledge, both to expand the system's knowledge base and to reach solutions to the problems at hand. There exists a number of types of reasoning ranging from natural forms, such as reasoning by analogy and by generalization, to logical and procedural reasoning.

When choosing a knowledge representation scheme, one must consider both the types of knowledge to be represented and the methods by which the knowledge will be used within the system. In addition, since each "use" is based on the products of the previous "use," the techniques implemented to "use" the knowledge require a common form of represented knowledge.

3.3 Declarative Representation Schemes

Declarative representation schemes represent the "world" as events, objects and the relationships existing among them. These schemes give what is often regarded as a static representation of the world (Barr and Feigenbaum 1981). Declarative representation schemes exist in, among others, two important forms: logic schemes and semantic networks (Mylopoulos 1980).

3.3.1 Logical Representation Schemes

Logical representation schemes are schemes which represent the "world" as a group of logical formulas. These logical formulas

are referred to as the atomic units of the scheme. It is through the manipulation of these atomic units, according to the rules of logic, that a system is able to make inferences and act knowledgeably. The two most common logical representations are propositions and predicates, manipulated through the use of propositional calculus and predicate calculus, respectively.

Propositions are sentences representing knowledge which are assigned a value of either true or false. These propositions cannot be broken down into individual components but must be evaluated as a complete sentence. Some examples of propositions are: the truck has four wheels, the poodle and the bird are orange, and eighteen minus five is thirteen. Propositional calculus concerns itself with the syntactic manipulation of propositions to prove whether various combinations of these propositions are true or false (Barr and Feigenbaum 1981).

Predicates are statements representing knowledge about specific objects or individuals which are assigned a value of either true or false (Barr and Feigenbaum 1981). The objects or individuals used in predicates are called arguments. Strings of these arguments connected by logical operators are referred to as well-formed formulas. Predicates are more powerful than propositions because the individual components of a predicate can be evaluated, i.e., a predicate can be broken down into its individual arguments. Some examples of predicates are: is blue (the sky), is equal to (3 + 4, 7), and is bigger than (an elephant, a mouse). Predicate calculus is the means for

manipulating predicates. In so doing, the truth values of the well-formed formulas can be determined. Predicate calculus is more powerful than propositional calculus because, in addition to having the same logic, it allows the use of quantifiers, such as "for all" and "there exists," and inference rules to introduce or eliminate these quantifiers (Barr and Feigenbaum 1981).

An extension of predicate calculus is first-order logic. First-order logic involves the use of functions which act similarly to predicates by representing relationships of objects but are not limited to true-false responses (Barr and Feigenbaum 1981). Functions can also respond by giving other related arguments which are represented within the system. In addition, first-order logic includes the use of the predicate "equals."

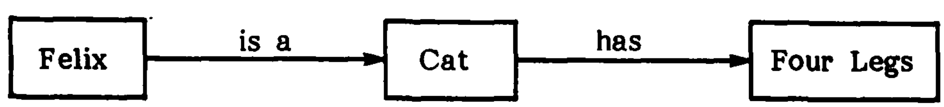
An important advantage of a logical representation scheme is the relatively natural and simplistic means of representing knowledge provided which will construct knowledge bases that can be easily understood in most instances (Mylopoulos 1980, Barr and Feigenbaum 1981). A logic scheme is also flexible, allowing single representation of facts (Barr and Feigenbaum 1981). Another strength of logic is the fact that it deals with precise, well understood and standard interpretations for the meaning of expressions (Mylopoulos 1980, Barr and Feigenbaum 1981). In addition, this scheme offers modularity which is achieved by allowing modification of the knowledge base through the introduction or deletion of logical formulas (Barr and Feigenbaum 1981). This can be done to supplement the knowledge

base without altering the validity of the system's deductions. One added advantage is that a number of inference rules are available for use in defining proof procedures (Mylopoulos 1980).

There are two major disadvantages associated with the use of logic schemes as a means of knowledge representation. The most important of these is the difficulty encountered when attempting to represent both procedural and heuristic knowledge through the same means when these two types of knowledge are used in two different ways (Barr and Feigenbaum 1981, Mylopoulos 1980). An additional difficulty, resulting from the lack of any structural organization principles, occurs when trying to manage the knowledge base as its size increases (Mylopoulos 1980).

3.3.2 Network Representation Schemes

Network representation schemes or semantic networks are also declarative representation schemes which model the "world" as objects and their respective relationships through the use of nodes and arrows (Mylopoulos 1980). The nodes represent objects within the world being modelled while the arrows represent relationships existing between the nodes which they connect. A network representation for the statement, a cat has four legs and Felix is a cat, might look as follows:



From this representation one can reason that Felix has four legs. These types of representation schemes have been found to be quite

useful in AI where the reasoning is based on taxonomy, or classification, and the properties of the objects being modelled (Barr and Feigenbaum 1981).

An important advantage of a semantic network representation scheme is that it is well suited for the retrieval process because of the access paths which are provided by the associations existing within the network. Another strength is that the knowledge base can be easily understood due to the ease with which it can be graphically represented. Additionally, the scheme has the potential for a somewhat structural organization, which can be done on the basis of the association types within the graphical representation (Mylopoulos 1980).

Mylopoulos (1980) mentions that a major disadvantage to semantic network schemes has developed due to the number of different applications of this scheme. As the various applications have developed, most also have developed terminology and semantics which are best suited for their needs. Semantic network schemes, however, lack a standard and final semantics. Another disadvantage which becomes obvious with large representations relates to the problems encountered when handling the vast number of integral objects, facts and relationships which result (Barr and Feigenbaum 1981).

3.4 Procedural Representation Schemes

Procedural representation schemes are a collection of procedures used to express the various aspects of the knowledge, both factual and heuristic, within the knowledge base (Barr and

Feigenbaum 1981). These procedures assist the system by directing how the knowledge base is to be used to make inferences and reach conclusions. Procedures directing the system attempt to follow lines of reasoning and alleviate the use of irrelevant knowledge (Barr and Feigenbaum 1981). Essentially a procedural representation scheme is the combination of declarative representations, of facts and objects and directions on how to apply or use the represented knowledge.

Procedural representation schemes are classified by the method which they use to activate procedures and by the control structures used to direct implementation of the system's knowledge (Mylopoulos 1980). The procedural schemes to be discussed are pattern directed inference systems and production systems. Of these two methods, the pattern directed or guided inference system most nearly fits the definition of a procedural representation scheme.

3.4.1 Pattern Directed Inference Systems

Many different pattern directed inference systems exist, each with their own peculiarities and applications. PLANNER, (Hewitt 1972) a well-known pattern directed inference system, will be used for this discussion. PLANNER is a programming language developed to effectively represent knowledge as well as supply the control information essential for the manipulation of this knowledge. The knowledge base of PLANNER consists of knowledge represented as assertions and theorems, which are referred to as "demons." The theorems and the patterns by which they are interrelated serve as the procedure by which knowledge is to be manipulated within or

added to the knowledge base. Each time data is manipulated within the data base an associated theorem is activated. If the theorem matches the data, the theorem is executed, thereby supplying a procedure for using the data. If a match does not exist, another theorem is activated and tested. This process continues until a match is found (Mylopoulos 1980). The theorems are not tested in any set order. These executed theorems give PLANNER instructions as to how other theorems are to be used, what other theorems may be applicable and the ability to suggest general classes of theorems.

Since PLANNER was developed as an AI programming language, it has been used for a number of AI research applications, the most extensive of which has been the SHRDLU system developed by Winograd in 1972. The major advantage of the representation scheme used by PLANNER is the ability it has to represent heuristic knowledge while implementing this knowledge to direct the manipulation processes (Barr and Feigenbaum 1981). This ability gives the system a more directed line of inference through the use of domain specific heuristics thus avoiding unnatural lines of reasoning. An additional advantage of a pattern directed inference system is its ability to update its data base, thereby keeping the available knowledge relevant to the situation at hand while giving the system the ability to model its current state (Barr and Feigenbaum 1981). In addition, the pattern directed inference system method of representation allows facts to interact directly because both factual and procedural knowledge act together.

Among the disadvantages encountered when using a procedural representation scheme such as PLANNER are the problems realized when incomplete knowledge exists (Barr and Feigenbaum 1981). Incompleteness occurs when the knowledge base does not contain enough information to prove important aspects of the world modelled (Levesque 1980). This will, in many cases, cause inconsistencies when reasoning with this knowledge. Another disadvantage is that there exists little modularity of knowledge (Barr and Feigenbaum 1981). Since the heuristic information forms a number of interactions between facts, any change or addition to the knowledge base may seriously affect the system's logic. The control information also presents a problem at times because facts coded within the control structure cannot be expressed without also expressing the related control information. In addition, the system is unable to reason with its control information and the knowledge base is not easily understood.

3.4.2 Production Systems

Production systems, which are another type of procedural representation scheme, are often regarded as a separate scheme of representation. Production systems consist of production rules, a data base and an interpreter (Davis and King 1977). Production rules, which form the knowledge base, are used to manipulate the data base. They are statements having a situation-action format which can be broken down into a left-hand side and a right-hand side. The left-hand side contains a description of some situation or condition which must exist in the data base in order to carry out

or "fire" the action or set of actions contained in the right-hand side of the rule. The data base, as the name implies, is the collection of facts and assertions used to represent the system's "world." The interpreter, or inference engine, decides for the system the order in which rules are to be applied. This decision is made on the basis of an evaluation of the present contents of the data base. Since the order in which rules are applied varies depending on the method of evaluating the data base, it can be seen that the interpreter is what gives each production system its uniqueness.

A production system operates in a cycle consisting of three phases: matching, conflict resolution and action (Barr and Feigenbaum 1981). The initial phase is the matching of production rules to the current data base, testing each rule according to a specified order to see if it matches (Mylopoulos 1980). If more than one rule matches, the interpreter must decide which rule to carry out. This operation is referred to as conflict resolution. Lastly, the action of the selected rule is carried out, changing the data base and thus creating a new situation. The cycle then begins again as new rules are matched against the new situation. Davis and King (1977), in their treatise on production systems, suggest three domains for which production systems are applicable. These include:

1. Systems where it is important to detect and deal with a large number of independent states
2. Systems which require a broad scope of attention and have the capability of reacting quickly to small changes

3. Systems where knowledge of the problem domain falls naturally into a sequence of independent "recognize-act" pairs

There exists a number of advantages to the knowledge representation scheme used in production systems. The two most important advantages offered by production systems are their modularity (the knowledge base is easily changed simply by adding, deleting or changing production rules) and their uniformity (all production rules are constructed using the same format) (Barr and Feigenbaum 1981). Other advantages include the ease with which production rules can be accessed (enabling exposure of the system's line of reasoning) (Davis and King 1977), the naturalness with which many types of knowledge can be expressed (Barr and Feigenbaum 1981), and a knowledge base which can be fairly easily understood (Mylopoulos 1980).

As with any system, disadvantages to the use of production systems also exist. Possibly the most significant disadvantage is the system's inherent inefficiency (Barr and Feigenbaum 1981). Inefficiency results from the step by step attempted match and performance of each rule in a predetermined sequence even when prior or additional knowledge could be applied to significantly reduce the number of steps. An additional disadvantage is that the formalism of production systems creates a flow of control which is hard to follow, resulting from the inherent complexity of the inference engine. For a broader description of the concept of production systems and their use in AI see Nilsson (1980); Davis, Buchanan and Shortliffe (1977).

3.5 Frame and Script Knowledge Representation Schemes

The additional knowledge representation scheme discussed here attempts to combine the various advantages of both procedural and declarative representation schemes. Frames, developed by Minsky, are data structures used to represent a stereotype of objects or concepts. Data structures are made up of slots which contain information about the various aspects of the situation being described and the relationships which exist between the slots (Mylopoulos 1980). Slots may also reference other frames (Quinlan 1980). Attached to each frame are instructions regarding how the information in the frame is to be used depending on the situation at hand (Mylopoulos 1980).

A frame system operates by taking a situation of the "world" and attempting to construct a description of it from the information contained in the system's data base. By decomposing the "world" situation and attempting to match these components to stored frames, which represent a nearly identical situation, a description is constructed. The final representation is a frame for each situation described containing the pertinent information about the "world" situation filled into the specified slots (Barr and Feigenbaum 1981).

The declarative nature of frame schemes is contained in the method used to represent individual facts pertaining to an object or concept. The procedural nature results from the use of instructions which are attached to the slots and frames to direct the implementation of the knowledge represented. The power of frames is realized when representing individual situations.

Scripts, on the other hand, are best suited for the representation of sequences of events. Scripts invented by Schank and Abelson can be thought of as sets of frames representing events (Barr and Feigenbaum 1981). These outer frames also have information attached to them specifying their use, giving the system the ability to use a series of event frames together to represent sequences of events.

An important reason for the development of the frame and script representation schemes was to combine some of the advantages of declarative representation schemes with the advantages of procedural representation schemes. The major advantage is the ability of the frame and script representation schemes not only to represent individual facts about a situation, in a declarative nature, but also to contain procedural information, which can be heuristic in nature, regarding the use of the knowledge represented (Barr and Feigenbaum 1981). Other advantages include the ability of the system to contain self-knowledge, a well-organized representation structure and default values which can be used to cope with incomplete knowledge (Barr and Feigenbaum 1981).

One of the disadvantages of the use of frame and script representation schemes is the problem of decreased modularity as the complexity of the knowledge represented increases, resulting from the increased number of interrelationships existing between and within frames. Another disadvantage is that frame systems cannot deal with a number of different hypotheses at the same time; they must work with one "best" hypothesis until a better one is discovered (Kuipers 1975, Barr and Feigenbaum 1981).

3.6 Conclusions

The means by which knowledge is represented within an AI system must be carefully chosen to allow the system the best possible use of this knowledge. Of the representation schemes mentioned, three basic aims are desired by each (Mylopoulos 1980): 1) to have the ability to construct a description of some "world," 2) to serve as the linguistic means for these descriptions, and 3) to give as "natural" a means of representation as possible.

The representation schemes discussed exhibit both advantages and disadvantages to their use. When selecting the right scheme, one must consider the types and structure of the knowledge to be represented as well as the desired attributes of the system's performance. Some additional considerations include the degree of understandability of the knowledge base, the flexibility of the knowledge base, the modularity of the knowledge base, the clarity of the line of reasoning, the completeness of the information available and the efficiency of the process.

Each operation of an AI system relies on the ability to use knowledge contained within the system. It is, therefore, essential that each aspect and use of the knowledge base be carefully considered before selecting a representation scheme.

CHAPTER IV

SEARCH

4.1 Definition and Overview

Search refers to the technique used by AI systems to systematically explore a network or graph structure in the process of working towards a goal contained within the structure. Search is also regarded as a tool for problem solving (Barr and Feigenbaum 1981), i.e., searching through a graph for a solution. A search system is composed of a data base, operators and a control strategy (Barr and Feigenbaum 1981). The data base is the collection of data or knowledge to be searched and the operators can be viewed as the rules used to manipulate the knowledge. These operators are either general or specific and can be used to generate other states if directed to do so. The control strategy refers to the method of searching the search space and often takes the form of an algorithm.

4.2 The Search Space

A search space is either an explicit or an implicit representation of the network or graph which contains the alternatives to be explored by the search process (Raphael 1976). An explicitly represented search space is a diagram or graph depicting the information in the data base as nodes and the system operators as arcs.

(Figure 4.1 is an example of an explicitly represented search space, in which the nodes are identified by the letters A through G.) An implicitly represented search space is the collection of operators or rules which can be used to generate the graph. The nodes of a graph represent states or knowledge of the problem being represented, while the arcs which can be viewed as operators represent the relationships existing between the states or nodes which the arcs connect (Barr and Feigenbaum 1981).

A variety of nodes can exist within a graph. A parent node or predecessor node is a node from which a number of nodes are generated and a successor node refers to a node which generates from another node. Additionally, terminal nodes are nodes which exist at the end of branches within the graph, not necessarily goal nodes, and goal nodes are nodes which are the target of the search. The rank of a node in a graph refers to the "cost" associated with achieving that node (Williams 1981a). Directed arcs represent one-way relationships existing between nodes while bidirectional arcs represent relationships between nodes which are capable of directly referencing each other. Nodes and arcs can be used to represent a wide variety of knowledge and operators. For example, the graph shown in Figure 4.1 could be used to represent a route location problem where the nodes represent locations, such as possible load or unload points, while the arcs represent the distance between locations. Raphael (1976) suggests the following four possible uses of nodes and arcs:

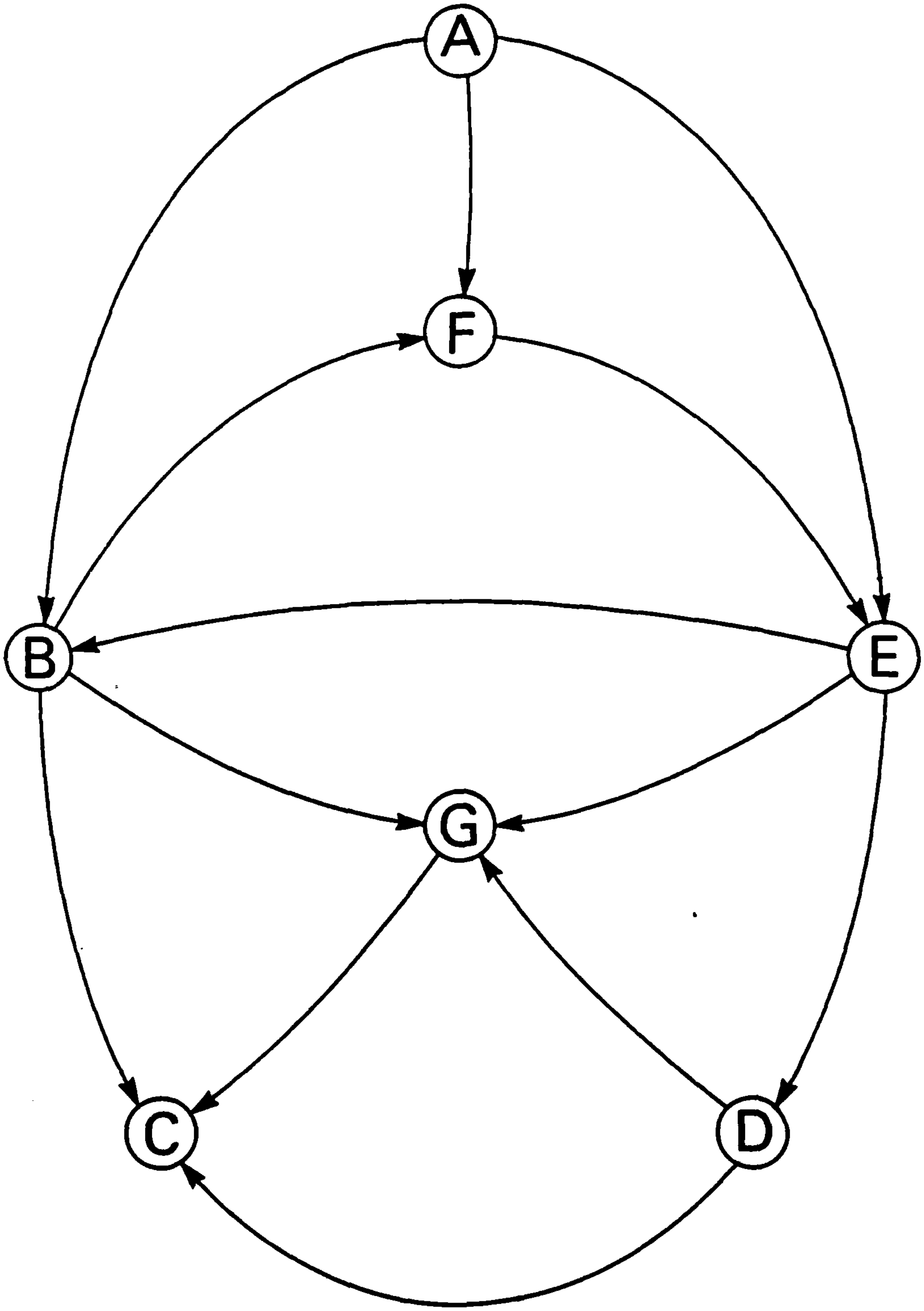


Figure 4.1 Directed Graph

1. Nodes—Represent candidate solutions to be tested
   Arcs—Represent rules for finding more nodes
2. Nodes—Represent tentative conclusions
   Arcs—Represent additional hypothesis
3. Nodes—Represent the state of a process or computation
   Arcs—Represent elementary actions used to change each state
4. Nodes—Represent complete descriptions of problems
   Arcs—Represent relationships existing between these problems

Graphs can be subdivided into a variety of types. The most common and one of the easiest to work with is the tree structure (see Figure 4.2). There are a number of characteristics which distinguish tree structures from graph structures (Raphael 1976). A tree structure is a special type of graph which has an initial node, referred to as the root node. The root node has no predecessors and is the node from which all successor nodes are generated. All nodes within the tree can have only one predecessor and with their corresponding arcs constitute the branches of the tree. The terminal nodes are referred to as the leaves of the tree. Although goal nodes can exist within the leaves of the tree, one or more of the leaves usually will be a goal node. Additionally, no loops can exist within the branches of a tree structure, whereas nodes in a graph structure can have successors of equal or lesser rank thereby forming loops (Williams 1981a).

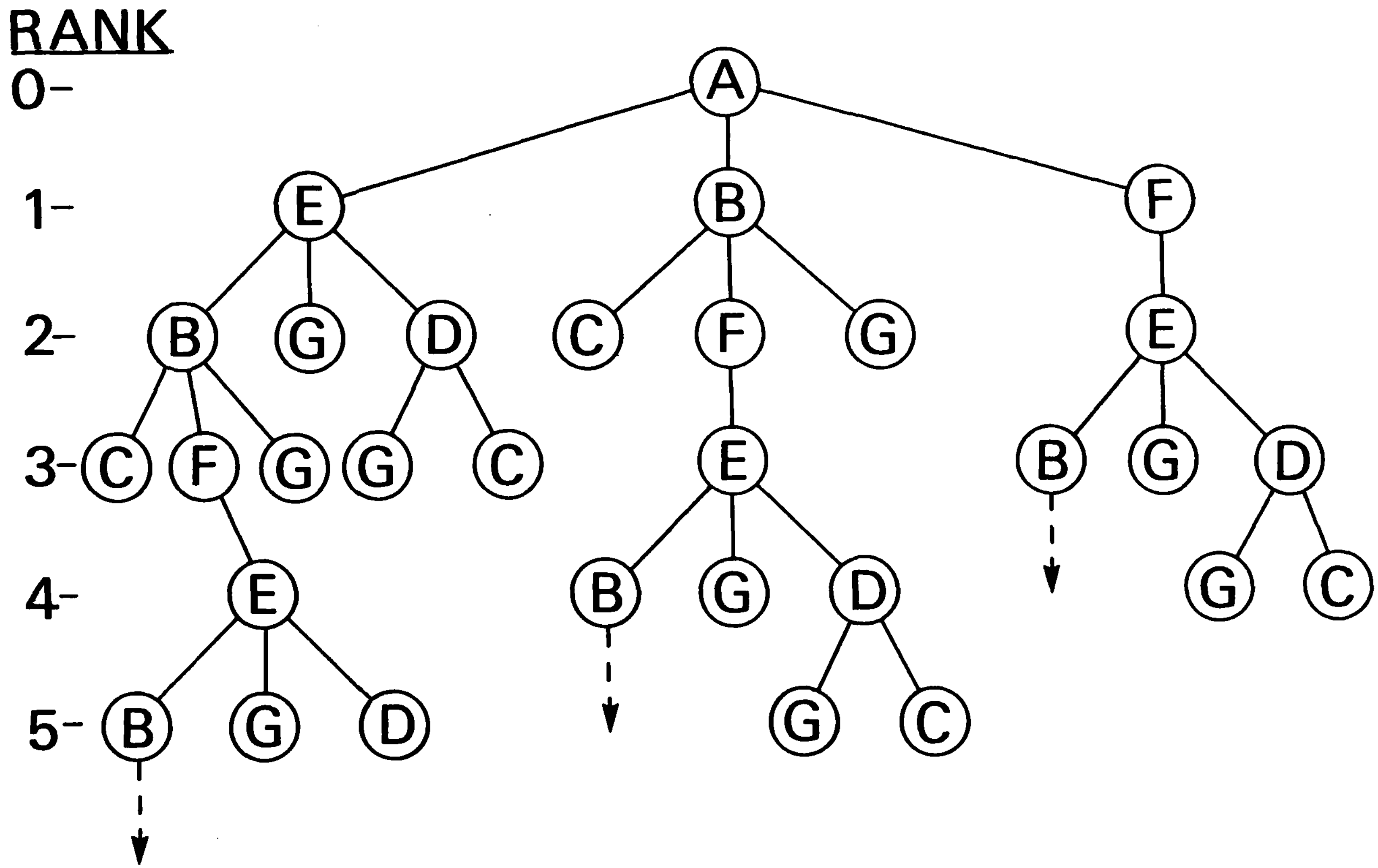


Figure 4.2 Tree Structure

Most graph structures can be represented as tree structures, e.g., the graph structure of Figure 4.1 has been represented as a tree structure in 4.2. However, when this is done, a number of problems can be created. When transforming graph structures containing loops, it becomes necessary to repeat the nodes forming the loop within the tree structure, e.g., this occurs with the nodes B, E & F which form a loop in the graph structure of Figure 4.1 but must be repeated in the tree structure of Figure 4.2. Additionally, each node within a graph can have many predecessors and the implicit representation of a graph containing a loop, such as the one shown in Figure 4.2, will create a tree of infinite depth.

4.3 The Control Strategy

The control strategy of each system is that component of a system which decides the order of search. It represents some systematic method of working through a graph structure. The control strategy governs which operators will be fired and in what order, thereby controlling what portions of an implicitly represented graph will be generated, i.e., determining what portions of the data base are to be exposed for searching. Among the number of existing control strategies, the two most common theories are forward reasoning and backward reasoning. These two theories represent a number of specific search strategies (Barr and Feigenbaum 1981).

Forward reasoning is the process of scanning a group of possible operators, choosing an applicable operator whose assertion matches that represented in the data base and applying it to update

the data base (Duda and Gaschnig 1981). This process continues until the situation represented by the data base satisfies a goal condition or until no more applicable rules exist. The theory of this strategy is to bring the data base from its initial state, through a series of changes until a goal condition has been satisfied (Barr and Feigenbaum 1981). This strategy is a "data driven" strategy and is referred to as antecedent reasoning or forward chaining. An example of forward reasoning would be the process used to simplify complex equations by manipulating the various arguments prior to actual solution.

Backward reasoning involves selecting a goal to be achieved and searching for the operators which, when fired, will bring the selected goal closer to the situation at hand (Barr and Feigenbaum 1981). The selected rules when applied create sub-goals which are simple subsets of the goal. The process continues until the sub-goals or sub-sub-goals can be easily solved and, when grouped together, will represent the initial situation. This strategy is regarded as "goal-driven" and is referred to as consequent reasoning or backward chaining. An example of the backward reasoning process would be the process used to solve puzzles. The desired picture is known yet one must search for specific pieces which, when in place, will reduce the size of the unconstructed portion.

Another form of control strategy is combination of forward reasoning and backward reasoning which is referred to as a difference directed or parallel strategy (Barr and Feigenbaum 1981). The difference directed strategy reasoning involves determining the differences which exist

between the present situation and a chosen goal state and then selecting and applying those operators which will most effectively reduce these differences. The selected operators can be applied to either the present situation or the goal situation which are represented in the data base (Nilsson 1980). The rules are selected in a fashion similar to that used by forward reasoning and backward reasoning. However, both processes occur simultaneously and the data base is used to give insight as to the differences which need to be eliminated. The selection process is repeated until the updated present situation is similar to the existing goal situation. Means-ends analysis, which will be discussed later in this chapter, is a technique using this type of control strategy where the comparison of the present situation to the goal situation is used to increase the efficiency of the selection of operators.

The selection of a control strategy should be based on the configuration of the search space and the type of description represented in the data base. Forward reasoning requires that the initial states be represented in the data base and is most efficient when used for graphs which have many intuitively clear initial states relative to only a few goal states. Forward reasoning also allows the storage of all trial paths during the search, giving the system the ability to resume work on any trial path should the present path prove inappropriate (Nilsson 1980).

Backward reasoning is best suited for graphs containing a number of goal nodes which require searching only a small amount

of space and should be searched in a tentative manner, i.e., if the current line of search is determined to be inappropriate it allows the system to try another one instead (Nilsson 1980). However, with backward reasoning only the current path being explored is stored. Backward reasoning requires that the goal states be represented in the data base.

Bidirectional reasoning or difference directed reasoning requires that both the initial states and goal states be represented in the data base. This type of strategy is well suited for problems with clearly defined initial states and goal states which require a tentative form of search.

4.4 Search Strategy: Definition and Overview

A search strategy is the process or series of steps which are followed to search a given search space. The search strategy is most often an algorithm which governs the order in which operators within the search space are to be fired, creating an explicit search space from an implicit representation of the search space. Search spaces can be thought of as specific types of control strategies. The search strategies first examined here will be those which are guaranteed to locate a solution through exhaustive search. Restrictions which can be imposed on these basic methods to improve their efficiency will then be discussed. Lastly, search strategies which selectively limit the size of the search space will be discussed. (The

search strategies are discussed here relative to their use in tree searches, allowing the emphasis to be placed on the techniques rather than the structure of the search space.)

4.5 Exhaustive Search Strategies

There are two basic search techniques from which many derivations arise: depth-first searches and breadth-first searches. Both, in their "pure" form, are referred to as exhaustive searches (Williams 1981a). Exhaustive searches are search strategies which are guaranteed to find a solution path; however, due to the nature of the techniques used the searches are quite inefficient. A depth-first strategy begins at the root of the tree and searches each branch as deeply into the tree as possible, beginning its search on the left-most branch of the tree. Once the search has progressed as far as it can go on the left-most branch of the tree without discovering a goal node, it backs up to the nearest unexplored branch and searches its left-most path (Winston 1977). Figure 4.3 shows the process of using a depth-first strategy to search the given tree. The process continues until a goal node is discovered.

The depth-first search strategy is most effective when the search space is known to be relatively small, with all paths of similar length. Due to the lack of information about the search space the process is quite inefficient. As a result, a tree with paths of unequal lengths may have a good portion of its lower branches explored before the goal node, which may be on a short path within the middle of the tree, is found. Additionally, the depth-first search strategy does

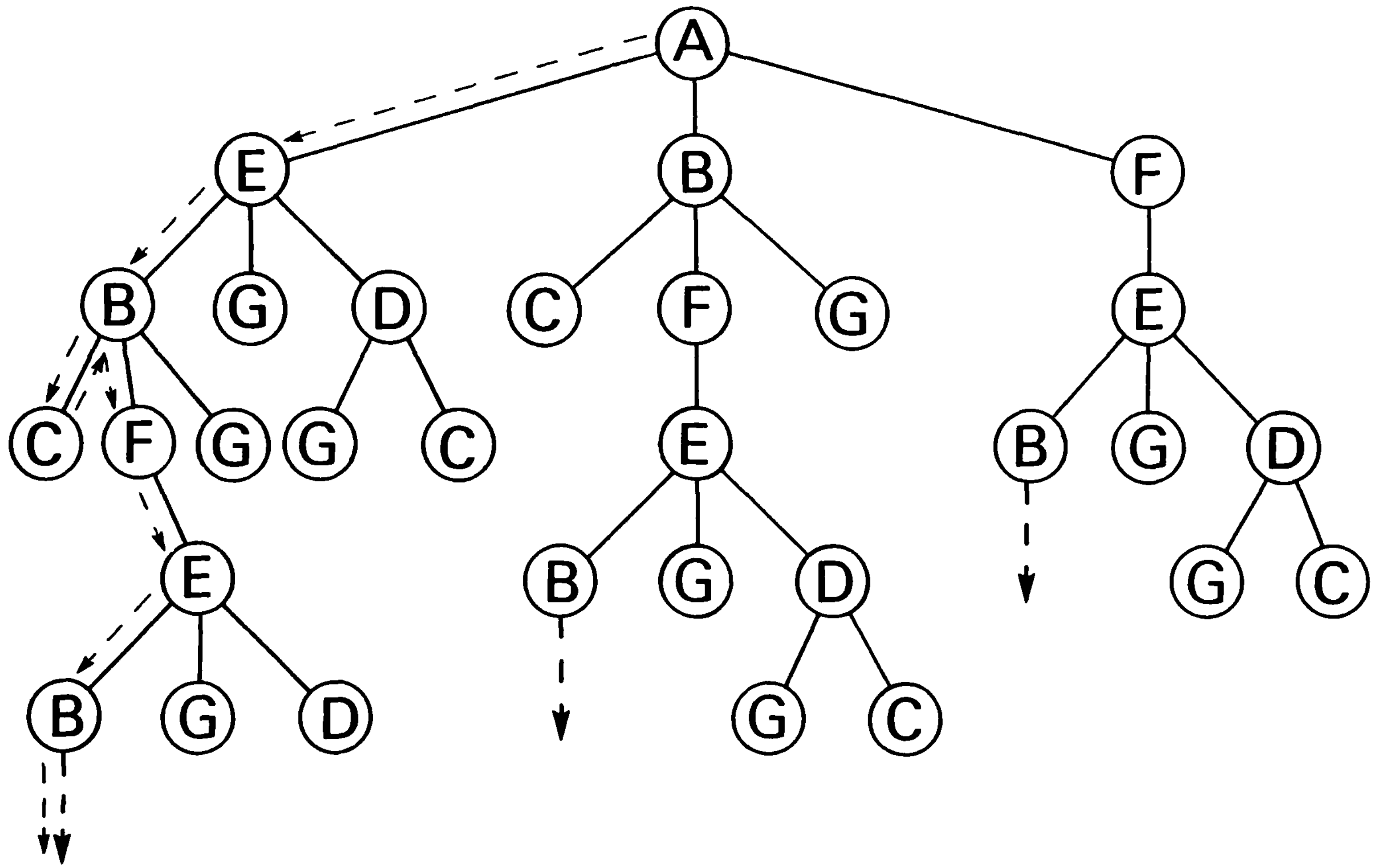


Figure 4.3 Depth-First Search

not guarantee finding an optimal path and, if an infinite branch is searched prior to locating a goal node, no solution will be found (Raphael 1976).

Breadth-first searches begin by examining the immediate successors of the root node, i.e., the first rank. If no goal node is located, the successors of these nodes are then examined, usually in a left to right fashion across the breadth of the tree. This process of inspecting each successive rank of the tree continues until a goal node has been located (Winston 1977). Figure 4.4 is an example of an application of the breadth-first search strategy. Breadth-first searches are guaranteed to find a solution and the goal node located will be the one having the minimum path length (Raphael 1976). This method, however, also proves to be inefficient, especially when all paths are of similar length, because of the lack of information about the search space.

Depth-first and breadth-first search techniques rely on the brute force of their strategy and can be regarded as the most basic forms of search. Their effectiveness is essentially dependent on the shape of the tree being searched and the location of the goal node within that tree. Both methods are applicable as a data driven search for which they are most commonly used. However, depending on their specific application, they can also be implemented as a goal driven search. The search strategies examined next essentially use the same methods or a combination of them, except they now have some form of restriction placed on the technique to reduce the size of the search by increasing the efficiency of the search strategy.

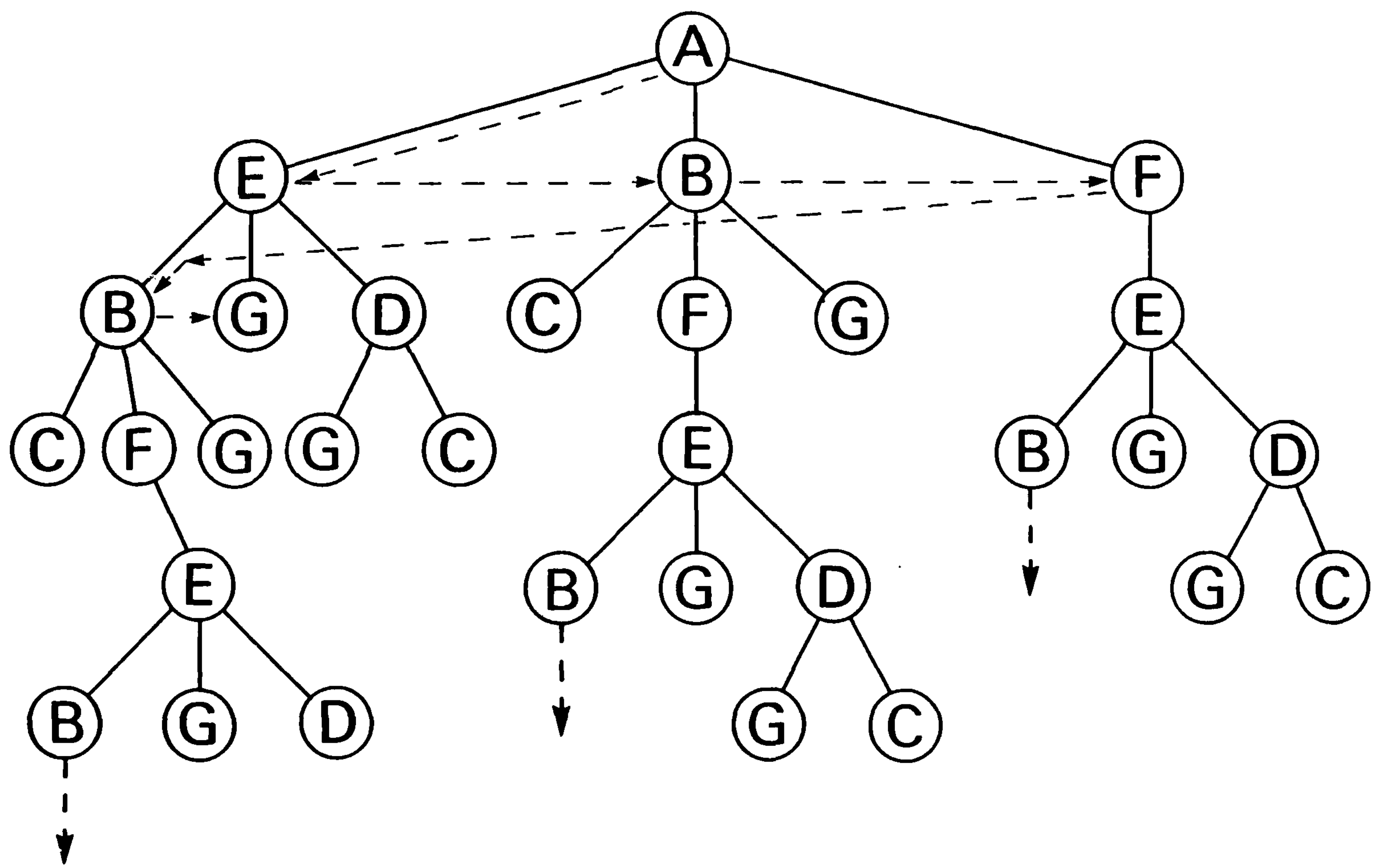


Figure 4.4 Breadth-First Search

4.6 Limited Depth-First Search

The limited depth-first search is a depth-first strategy which is allowed to progressively deepen its search into the tree (Raphael 1976). The search begins as a depth-first search; however, the depth that is searched is limited or bounded. After reaching this depth on the left-most branch the system then begins backtracking and checking each node progressively through the breadth of the tree up to the specified depth. If this portion of the tree has been exhaustively searched and no goal nodes have been found, the depth is extended and the process is repeated until a goal node is located. Figure 4.5 shows how a tree would be searched using a limited depth-first strategy.

The limited depth-first method is guaranteed to find a solution and will avoid searching the lower portions of a tree unnecessarily. Additionally, if the solution located is not the solution of shortest path length, it will lie in the same level or rank of the tree as the solution of shortest path length. This search strategy, however, does not guarantee to find the optimal solution and is inefficient because of the lack of information about the search space.

The search strategies previously mentioned are referred to as blind searches (Barr and Feigenbaum 1981). A blind search is a search strategy which uses no domain specific information to assist the decision making process when selecting the next alternative. The choices are strictly arbitrary, based solely on the order specified by the algorithm, often resulting in the creation of huge search spaces.

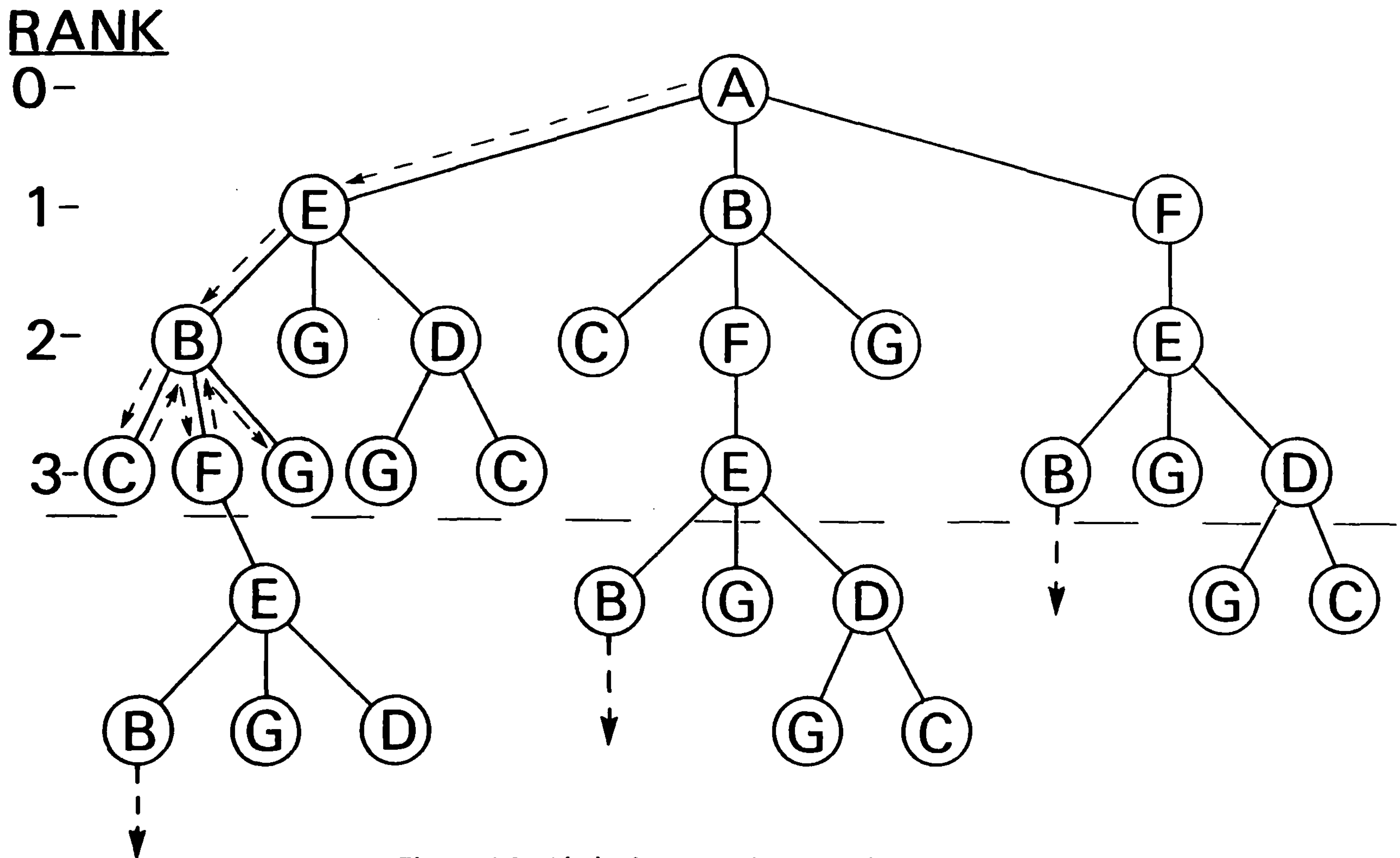


Figure 4.5 Limited Depth-First Search
Depth of search is limited to 3 ranks

A search strategy using additional information about the alternatives to limit the size of the search and increase the efficiency of the search is, among others, a heuristic search. A heuristic search accomplishes this by using the heuristic information to determine which node should be expanded next, which successors of each expanded node should be generated and which nodes should be discarded (Barr and Feigenbaum 1981).

4.7 Heuristic Search Strategies

The most basic type of heuristic searches are ordered searches which essentially are depth-first or breadth-first searches with heuristics applied to assist in selecting the best node from the alternatives. The heuristic value associated with the searches discussed here is referred to as an evaluation function (f*) and can be based on any one of a number of parameters or estimates (Nilsson 1980). The smaller the value of the evaluation function the better that alternative will be. In their purest forms evaluation functions represent a value based on the rank of the particular node to which it is assigned. Another common value for f* to represent is the cost associated with choosing a certain path. The values of f* can be assigned to either the arcs or the nodes of the tree. Heuristic search strategies which use f* to determine a minimum cost solution are referred to as admissible algorithms (Barr and Feigenbaum 1981). The following discussions deal with search strategies which implement these heuristic techniques to assist the search process.

4.7.1 Ordered Depth-First Search

An ordered depth-first search is a strategy in which an heuristic value is associated with each decision point to provide a means for the system to choose the next alternative to be explored. The system begins at some initial node or root by opening each of its immediate successors. Each of these alternatives is then assigned an evaluation function. The value for f* is based on an evaluation of some stated characteristics or estimates, often an estimate of the relationship existing between that node and the goal node (or root node), i.e., the estimated distance to the goal node. The system then chooses the alternative with the f* of minimum value, providing the best overall improvement to the present situation. The process then repeats itself until a goal node has been discovered (Barr and Feigenbaum 1981).

The main advantage to this type of search is the increased efficiency realized by using a means of evaluating alternatives based on domain specific knowledge, thus reducing the number of nodes which will be searched prior to locating a goal (Barr and Feigenbaum 1981). The greatest disadvantage to this method stems from the fact that the strategy is only selecting a best local alternative rather than a best global alternative. Thus it does not guarantee finding the optimal path. The problems this causes arise when the system is forced to choose an alternative from choices which do not include the highest global choice or when the system must decide between alternatives of the same value. Additionally, ordered depth-first

searches become complex and often difficult to implement on multi-dimensional searches (Winston 1977), i.e., a search whose evaluation function is based on a large number of considerations.

Two variations of the ordered depth-first strategy exist: a minimum cost search (Raphael 1976) and a best-first heuristic search (Winston 1977). The best search technique for finding the path of minimum cost, based solely on the information contained within the tree, is the algorithm for trees (A^T) (Raphael 1976). This method is similar to the ordered depth-first strategy described previously except for the evaluation function. In this strategy each arc has a cost associated with it. Costs are assigned when the tree is constructed and the evaluation function of each node is the sum of all the costs along the path leading from the initial node to that node. The process used for searching the resulting tree is the same as that used by the ordered depth-first strategy. This strategy guarantees the discovery of the path of minimum cost and does it in the most efficient manner, i.e., opening the fewest possible nodes.

Figure 4.6 offers an example of how an ordered depth-first search using costs as the evaluation functions would work. The problem is to determine the fastest route from node A to node G. The values assigned to the arcs are the times associated with travelling that segment of the route. The cost for travelling the chosen route would include the equipment and labor costs per unit of time multiplied by the resulting time. It should be noted, however, that this method is not a "pure" heuristic search since the costs associated with the

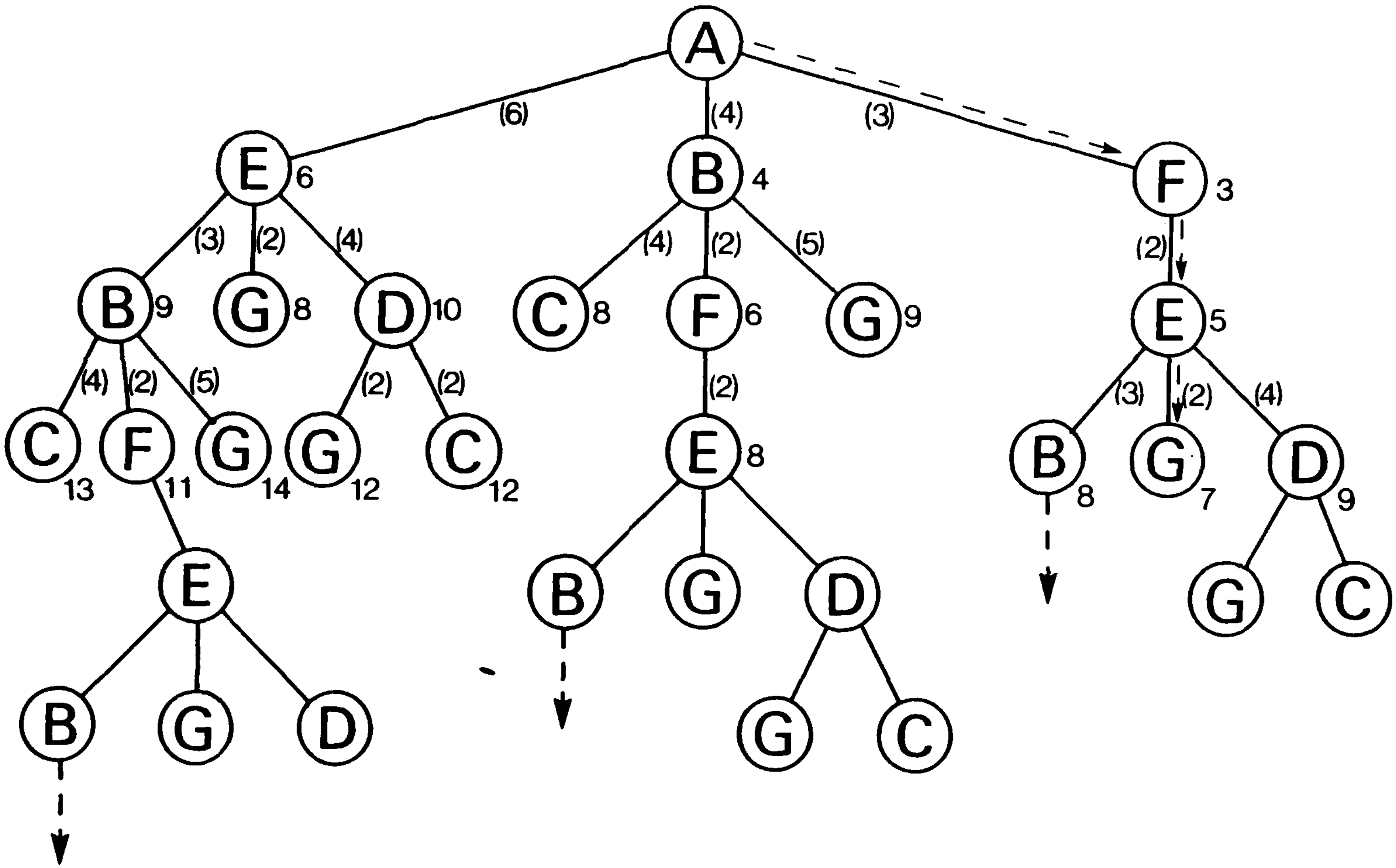


Figure 4.6 Ordered Depth-First Search
Using cost as the evaluation function

arcs are not estimates but actual values which have been assigned to the arcs prior to searching the tree.

A second variation of the ordered depth-first search which is often more effective is the best-first heuristic search. This strategy is the same as that used by the ordered depth-first search. However, when choosing a best alternative to expand, the evaluation functions of all open nodes, regardless of their position within the tree, are compared (Winston 1977). The alternative which is selected is then a best global alternative, thereby eliminating many of the problems encountered when pursuing the best local alternative. This gives an efficient search while locating a path which approaches the optimal one. For additional information see the discussion of the B* tree search algorithm by Berliner (1979).

4.7.2 A* Algorithm

An A* algorithm is a search strategy which maximizes search efficiency and guarantees locating the path of minimum cost (Nilsson 1971). This strategy assigns an evaluation function $f^*(n)$ to a node based on the cost $g^*(n)$ of the path from that node (n) back to the initial node added to a heuristic value $h^*(n)$ which is an estimate of the cost of the path from that node (n) to the goal node, i.e., $f^*(n) = g^*(n) + h^*(n)$. In this evaluation function $g^*(n)$ would be assigned an actual value and $h^*(n)$ would be assigned an heuristic value. Both values should be non-negative and the heuristic value should be an estimate which is as near to the actual as possible without exceeding it. The process of determining the path to a goal

node is identical to that described in the discussion of the ordered depth-first search with the exception that now the evaluation function is a summation of an actual value and an heuristic value.

Two important properties of this strategy are its admissibility and its optimality. The admissibility of the algorithm refers to its ability to find the optimal path to a goal node, if one exists (Nilsson 1977). The optimality of an algorithm refers to the closeness of the chosen path to being the "best" or optimal path existing within the tree, this property being dependent on the accuracy of the heuristic values h*(n) (Barr and Feigenbaum 1981). The A* algorithm is guaranteed to give the minimum cost path to a goal node provided that the heuristic value is always positive and less than the actual value it estimates. In addition, this strategy also gives an optimal solution since it expands the fewest possible nodes in the process of locating a goal node (Raphael 1976).

There are a number of variations of this type of heuristic search, the most common of which is a specialized version of the A* algorithm for use in tree searches referred to by Raphael (1976) as the A^{KT} algorithm. Another variation, called the branch and bound strategy, is discussed by Winston (1977). It should be noted that, when using the various types of searches which employ an evaluation function the effectiveness of the search will be dependent on what parameters are used to calculate a value for f*. It is therefore important to choose carefully those parameters which will best represent the current state of the search throughout the process.

4.7.3 Parallel Search

Parallel searches are searches which simultaneously search multiple segments of a graph or a tree. Bidirectional search, discussed in this section, is a special type of parallel search which implements both forward and backward reasoning and is carried out in opposite directions from two distinct points in the graph, i.e., each search process working toward the other (Nilsson 1980). The search is carried out by simultaneously generating two graphs, one originating from the initial node and the other originating from a goal node. As the graphs are generated simultaneously the corresponding states are compared to determine if any nodes have been identified by both searches. If a common node or state exists, the search is terminated and a path is located from the common node to both the initial and the goal node.

The strategy used to carry out this search can be any of those previously mentioned. However, the tree must be suited for both forward and backward searches, i.e., one must be able to clearly define initial states as well as goal states prior to the search (Barr and Feigenbaum 1981). The major advantage to implementing this type of search is the amount of effort which can be saved by developing two smaller graphs which will cease generation when joined rather than developing one graph which may grow quite large, possibly searching a number of terminal nodes prior to discovering a goal node (Nilsson 1980).

4.7.4 AND/OR Graph Search

Before discussing the strategy for searching an AND/OR graph it is necessary to describe the problem representation which uses this type of search strategy. Up until now all of the searches described are implemented on trees which are state-space representations of the problem, i.e., nodes represent the states of the problem while arcs represent the operators which are used to generate the next state of the problem from the current state (Williams 1981b). The AND/OR graph, however, is most often a problem-reduction representation graph. The problem-reduction representation uses nodes to represent problem or sub-problem descriptions while the arcs represent operators which can be used to transform the problems into sub-problems (Barr and Feigenbaum 1981). The concept involved is to allow the search to continue until the problems have been reduced to sub-problems which can be easily solved (Raphael 1976).

An AND/OR graph is a special type of problem-reduction in which the nodes are classified as either AND nodes or OR nodes. An AND node is used if the sub-problems created are a set of simpler problems which together are equivalent to the problem described in the parent node, i.e., all of the sub-problems must be solved to get a solution for their parent node. The OR node is used to represent problems which are simpler problems, each equivalent to the parent problem (Barr and Feigenbaum 1981). Pure OR graphs or pure AND graphs seldom exist. Instead they are often intermixed within the graph, representing how the problem can be transformed into sets of sub-problems or equivalent problems. AND/OR graphs are often

thought of as a means for solving problems, accomplished by searching the possible transformations and simplifications of the given problem for the best solution. Although this technique for representing problems is often a very natural way to represent a problem, the resulting AND/OR graphs may become quite large and complex when dealing with difficult problems (Winston 1977). Since the search strategies for AND/OR graphs are quite similar in concept to those discussed earlier, they will not be dealt with here. For additional information on this topic see Nilsson (1971) and Barr and Feigenbaum (1981).

One additional type of search strategy should be mentioned at this point since it is also used for searching problem-reduction representations. Means-ends analysis is a type of bidirectional search of problem-reduction representations which involve the generation of a graph from the current state guided by the differences existing between that state and the desired goal state. This strategy selects the operators which will have the greatest effect on reducing the existing differences (Winston 1977). The selection of operators is done in three main ways (Barr and Feigenbaum 1981): 1) transform the current state to the goal state through a series of transformations, 2) reduce the existing differences by modifying the current state, and 3) apply an operator to the current state to develope a new and less different current state. The result of these three strategies, when applied to a problem, is the generation of an AND/OR graph which will contain the solution. This type of search strategy was developed

for use in the General Problem Solver (GPS) by Newell, Simon and Shaw in 1957, and will be discussed in chapter five.

4.7.5 Game Tree Search

Game trees are an additional representation for which search strategies have been developed. A game representation differs from state-space representation and, to a lesser extent, from problem-reduction representation in that it represents all the possible states of the game and all the moves of the two opposing players from the point of view of only one of the players (Barr and Feigenbaum 1981). The search strategies, when using this type of representation, must be able to determine a winning strategy by evaluating the alternatives of both players. Therefore, the search must include a means of determining the most likely move of the opponent. Many of the searches which have been previously discussed can be altered to allow application to game trees and will not be discussed here. Additionally, games can be represented in the form of an AND/OR graph allowing the use of AND/OR search strategies. The two methods to be discussed which have been developed specifically for use in searching game trees are the minimax procedure and the alpha-beta pruning technique.

In a minimax procedure the strategy is to find the move of maximum value for the first player while assigning the move of minimum value to the second player (Nilsson 1971). The concept involved is that a "best" move for the first player will be the "worst"

move for his opponent and vice-versa. There are a number of variations of the minimax procedure: the one described in this section assumes partial generation of the tree prior to the search process.

The search begins at an initial node and generates, "looks ahead," a specified depth or number of ranks of the game tree. An evaluation function is then applied to generate the static value of each of these "terminal" nodes. The static value represents how favorable the "terminal" position would be for a win by that player. The system then backs up from these nodes assigning the maximum value of the successors to all the parent nodes (MAX) in odd-numbered ranks and assigning the minimum value of the successors to the parent nodes (MIN) in the even-numbered ranks (assuming the initial node of the tree is of rank zero). Once all nodes have been assigned the "backed-up" values, the first player (MAX) would select the node of maximum value for his first move. The "logical" second player (MIN) would then select the node of minimum value as his next move. The selection process continues in this fashion until the specified depth at which evaluations begin is reached. The process of evaluation and selection would then begin again and the search would continue until one of the players wins (Nilsson 1971).

The major advantage to this strategy is that it assigns some value to each possible position providing a criteria for the selection of each move. This method, however, has two major disadvantages: 1) the expense, in terms of time and effort, of generating all possible paths and calculating the static value of these paths, and 2) the static evaluation function is not an efficient means of representing

all the factors to be considered when selecting the next move (Winston 1977). A variation of this strategy is referred to as the negmax strategy, which is briefly discussed by Knuth and Moore (1975).

The alpha-beta pruning technique is a strategy which uses the concepts of static evaluation functions and "backed-up" values, as discussed under the minimax strategy, but significantly reduces the size of the tree which is generated and the number of static evaluation functions which are calculated by following only those paths which are most promising (Winston 1977). This procedure operates by simultaneously generating the "backed-up" values and alpha-beta values for the MAX and MIN nodes. The alpha values represent the "current largest final backed-up value" (Nilsson 1980) of the successors of a MAX node, while the beta values represent the "current smallest final backed-up value" (Nilson 1980) of the successors of a MIN node.

The procedure begins by generating a path to a terminal node and assigning static values to it. The system then backs up the path just generated, assigning alpha values to MAX nodes and beta values to MIN nodes. At each of these parent nodes the successors are checked to determine if any bad moves exist from that parent node, i.e., could the opponent force a play along that route which is worse than the existing opportunities. If so, it is unnecessary to continue searching the additional choices from that node. A potential bad move is encountered when a play could be forced in which a MAX alpha value would be less than the overall

best value which could be forced. A logical opponent is assumed to seek the moves which force play towards achieving the smallest value of beta. This process of assigning and selecting the optimal alpha and beta values continues until the initial node is reached. At this time an optimal path has been located from the initial node to a terminal node, which will have the same static value (Nilsson 1980; Winston 1977).

The most significant advantage to using the alpha-beta pruning technique is the amount of effort which is saved in developing the search tree while still allowing the strategy to locate the optimal path (Winston 1977). However, it is important to point out that the amount of effort saved by using this pruning technique is a function of the characteristics of the tree structure. Illustrations of the application of alpha-beta pruning to game trees can be found in Knuth and Moore (1975), Winston (1977) and Nilsson (1980). Winston (1977) also suggests a number of heuristic pruning techniques which can be used in conjunction with alpha-beta pruning to increase its effectiveness.

4.8 Summary

In this chapter a number of search strategies used by AI systems were discussed. Each strategy was best suited for a certain type of search space. The three types of search space mentioned were state-space representations, problem-reduction representations and game representations. One should also keep in mind that these representations can be either graph or tree structures which can be

either implicitly or explicitly represented. The search strategies discussed could also be categorized according to the method of search. Some used an exhaustive brute force method, e.g., depth-first and breadth-first, while others used varying amounts of heuristics and controls to direct the search, e.g., A* algorithm.

From this discussion it is apparent that, when selecting a search procedure, consideration must not only be given to the type of knowledge being searched and the means by which it has been represented but also to the effectiveness of the chosen strategy for the given search space. Many search procedures are quite powerful and when applied to an appropriate knowledge base can result in an inexpensive yet efficient means of search.

CHAPTER V

PROBLEM SOLVING

5.1 Definition and Overview

Problem solving is that technique of AI systems which gives them the ability to process information about situations or problems and determine a solution based on that information. The problem-solving process involves the previously described techniques of knowledge representation and search (Uhr 1973), i.e., the problem must be represented in a form which allows manipulation of the information and the solution process often involves a search of the viable courses of action to be followed. The methods used to represent the knowledge and to search the data base will affect the method by which a problem may be solved. AI problem-solving methods attempt to follow natural or human lines of reasoning to solve problems at the object-level and at the meta-level. Object-level problems are those problems which one encounters when dealing with knowledge about objects and physical entities, e.g., 4+3=?. Meta-level problems, on the other hand, are the problems of controlling the knowledge about what is known by the system, or self-knowledge, e.g., determining the best method of pruning large amounts of data.

The two most common forms of natural reasoning which have been implemented by AI problem solvers are derivation and

recognition (Raphael 1976). Derivation techniques of problem solving follow a sequence of steps which systematically guide the process of selecting a solution to the given problem. Derivation can be further divided into deduction and induction (Raphael 1976) where deduction is the technique of solving a problem by proceeding from general truths to specific truths and induction is the technique of solving a problem by proceeding from specific truths to general truths. Deduction is the most common method used in AI derivations: however, induction is used most often where rules are being formulated from examples. These techniques involve either formal reasoning processes, such as mathematical logic which follows precise steps according to the rules of logic, or informal reasoning processes, such as means-ends analysis which uses a difference-directed means of proceeding towards a solution (Raphael 1976).

Recognition techniques concentrate on being able to recognize a solution when it is discovered by the search process. This method of problem solving attempts to identify the situation at hand, usually some shape or figure, by matching it with patterns represented within the data base. This type of problem-solving is more commonly referred to as pattern recognition or pattern classification. There are two main techniques used for pattern recognition: discriminant recognition and syntactic recognition (Glorioso and Colon Osorio 1980). Discriminant recognition creates a set of points in a vector space to represent important features and characteristics of the pattern to be identified. The system then attempts to match these with patterns represented in a similar manner within the data base.

Syntactic pattern recognition, also referred to as structural pattern recognition, breaks a complex pattern down into a set of simpler patterns and then attempts to identify these simpler patterns. The initial pattern is then represented as a set of identifiable sub-patterns.

Specific problem-solving methods which implement the techniques of derivation and recognition are examined more closely in the following discussion. The methods are discussed in relation to their application to object-level knowledge. Finally, some of the methods which have been used to solve problems encountered when dealing with meta-level knowledge will be briefly mentioned.

5.2 Deduction

Deduction has been defined in the previous section as the technique of solving a problem by proceeding from general truths to specific truths. In AI systems the general truths are premises while the goal states are regarded as the specific truths. When developing a deductive problem solver there are three basic problems which must be overcome (Henschen 1976). 1) A suitable representation must be found, i.e., it must be easily implemented by the system. 2) Inference rules which can effectively manipulate the knowledge must be found. 3) The rules identified must be refined to some degree such that they locate proofs efficiently, i.e., the rules should use the least possible amount of time and memory when locating a proof.

There are a number of proofs used for deductive reasoning. The most widely known are those which apply mathematical logic

to prove a goal statement. These proof procedures adhere to a rigid structure or format and are classified as formal proofs (Raphael 1976). Proof procedures which do not apply only to a specific type of problem and do not follow a rigid structure are often regarded as informal proofs (Raphael 1976). The following section on deductive problem solving techniques briefly describes some proof procedures which use logics for reasoning and then some which implement informal techniques for reasoning.

5.2.1 Logic Systems

Logic systems implement a type of formal reasoning. They consist of a structured method of representing the initial situation as a number of premises and a set of inference rules. The inference rules are used to manipulate the premises in such a fashion that the solution or goal will be shown to be the obvious result. The basic means of representation for most logic systems is the well-formed-formula (wff), the most basic form being strings of symbols arranged in accordance with the grammatical rules of logic (Raphael 1976). Wffs represent individual concepts. However, by combining them into sentences and phrases the wffs can represent more complex situations.

A problem to be solved using logic must contain the given information which is believed to be true, in the form of premises, and a statement to be proven, referred to as a theorem (Raphael 1976). The system manipulates the premises by a selected set of rules attempting to show that the theorem is guaranteed to be true

if the premises hold true. A proof for the theorem is represented by the series of steps which result from the manipulation of the initial premises. The procedure or set of rules used to construct a proof for a specific problem will in part depend on the method used to represent the premises and the theorem.

The two basic classifications of proof procedures are semantic and syntactic (Raphael 1976). Semantic procedures are concerned with the specific meaning of the symbols in the wffs and reason by considering all possible interpretations of the wffs. Syntactic procedures, however, ignore the precise meanings of the symbols and reason by manipulating the wffs to create new ones which lead to the proof. The syntactic proof procedures are easier to implement on AI systems because they can be applied in a "mechanical way" (Raphael 1976).

Logical reasoning systems can be classified as one of two types: monotonic reasoning systems or as non-monotonic reasoning systems. Monotonic reasoning is a formal reasoning process which uses a strict or rigid logical system. This method of reasoning constructs proofs in such a way that each preliminary conclusion in the proof can be proven on the basis of the previous premises. Non-monotonic reasoning, an informal reasoning process, uses some of the formalities of monotonic reasoning while at the same time working with incomplete knowledge, i.e., a premise need not be provable when initially used but is merely assumed true. Additionally, in non-monotonic reasoning systems, if a better premise is found it can be substituted

for the initial one, thereby invalidating the initial premise (McDermott and Doyle 1980). Since this process of reasoning is quite similar to everyday human reasoning processes, it is often referred to as "common sense" reasoning (Winograd 1980). Non-monotonic reasoning will be discussed in greater detail in Section 5.2.5 of this chapter.

5.2.2 Formal Reasoning: Monotonic Reasoning

Monotonic reasoning procedures use logics which require that in order for the theorem to be true all the premises on which it is based must be true; therefore, all steps within the proof are provably true. The two methods most commonly used by AI systems for monotonic reasoning are propositional calculus and predicate calculus. It is important to note that although propositional and predicate calculus are discussed as methods for monotonic reasoning they can also be used for non-monotonic reasoning processes. In the following discussion propositional calculus is described along with two proof procedures, truth tables and Wang's algorithm. Predicate calculus and its principle proof procedure, resolution, are then described. When reading the discussion of proof by resolution, it should be kept in mind that with a few slight changes the resolution proof procedure can also be applied to propositional calculus (Raphael 1976).

5.2.3 Propositional Calculus

Propositional calculus, also known as sentential calculus, is a means of representing concepts through a combination of propositional variables and logical connectives. This representation allows

these concepts to be manipulated by the rules of a proof procedure. Propositional variables can be regarded as the most basic wffs of propositional calculus and are used to represent propositions (Raphael 1976). Propositions are declarative statements regarding a situation that have a known value of either true or false (Nilsson 1980). Propositions can be classified as either general or specific (Glorioso and Colon Osorio 1980). A general proposition makes a statement about a class or group of entities, e.g., the students went skiing, while a specific proposition is a statement about a single component of a class or group, e.g., Steve did not go to school today.

Logical connectives or propositional connectives are operators which are used to join single propositions or elementary wffs together to construct more complex wffs (Raphael 1976). These more complex wffs are referred to as sentences and take on a truth value determined by the truth values of the component propositions and their connectives. Additionally, these combined wffs are referred to as conjuntions if the logical connective is an "AND," and as disjunctions if the logical connective is an "OR." The more common logical connectives used in propositional calculus are listed with their meanings in Table 5.1.

Most of the previous section dealt with the format and methods of representation of premises and theorems. However, the problem solving ability exists in the proof procedure. At this point it is apparent that the representation is a crucial part of the problem-solving method for without a strict representation format of this type systematic proof could not be implemented. Before proceeding it is necessary to point out a few disadvantages of propositional calculus.

1) Propositional calculus lacks the ability to represent an "If. . . then. . ." causal situation even though it has the ability to represent an "If . . . then . . ." implies situation (Raphael 1976). This results from the relationship which exists within a causation, i.e., if an event occurs, causing a second event to occur, the second event is a result

Table 5.1. Symbols and Their Meanings in Propositional Calculus

Symbol	Is Read	Meaning
$\bar{p}$	Not p	p is false
$p \wedge q$	p and q	The intersection of p and q, common to both p and q
$p \vee q$	p or q	The union of p and q, either p or q or both
$p \supset q$	p implies q	If p is true, then q is true
$p \equiv q$	p iff q	p is true and only if q is true; p is equivalent to q

SOURCE: Glorioso and Colon Osorio (1980), table 3.1

of the first. An implied relationship is where the occurrence of one event indicates the possibility of a second event occurring, i.e., the second event is not a guaranteed result of the first event. No logical connective exists to represent a causal relationship, while a logical connective is shown in Table 5.1 which represents an implied relationship. 2) The smallest element of propositional calculus is the proposition. This allows no logical reasoning with components of the proposition (Raphael 1976). 3) Additionally, propositional calculus does not allow representation of elements of an undecided truth value, i.e., a variable such as x cannot be represented (Nilsson

1980). It should be noted, however, that propositional calculus is quite appropriate for simple domains. The proof procedures, discussed in the remainder of this section, are the most common for propositional calculus. Resolution is also a viable proof procedure; however, it will be discussed in relation to its use in predicate calculus.

Truth tables are semantic proof procedures which attempt to examine all possible combinations of propositional variables and the corresponding truth values of the resulting premises (Raphael 1976). A theorem is said to be true or proven only for situations where propositional variables have been assigned some combination of truth values which cause all resulting premises to be true. This procedure is guaranteed to give the correct answer. However, it tends to grow complex as the number of variables (2^n representations for n variables) and premises increase. Even though this system can be implemented on a computer, the system will be very inefficient. For a more complete treatment of truth tables as a means of proof for propositional calculus see Barr and Feigenbaum (1981) and Raphael (1976).

Wang's Algorithm is a syntactic proof procedure for propositional calculus which manipulates the propositional sentences producing the same result as truth tables, however, with much greater efficiency (Raphael 1976). The algorithm uses a syntactic procedure to manipulate the statements which can be used efficiently by AI sytems. The process as outlined by Raphael (1976) essentially involves the repetition of a series of manipulations used to create new wffs, thereby constructing new lines in the proof. Each of these new

lines must then be proven true. For this to occur there must exist the same elementary wff on both sides of the line, i.e., both sides of the equation. If all of the newly formed lines can be proven true, the theorem is said to be true or "proven." This type of proof can be easily applied to AI systems because of its straightforward structure and set of rules. To restate briefly, the major advantage of Wang's Algorithm is that it is probably the most efficient means of constructing a proof, which guarantees the correct answer, in propositional calculus. For additional information regarding this proof procedure see Raphael (1976).

5.2.4 Predicate Calculus

Predicate calculus is a method of logical reasoning which allows a more detailed representation of premises and theorems than propositional calculus. Predicate calculus involves all of the basic structure and format of propositional calculus while also including a number of properties which greatly increase its power and application. The predicate calculus discussed here is often referred to as "first-order" predicate calculus (Nilsson 1980).

Before discussing predicate calculus further it is important to bring out the two most significant differences which exist between propositional calculus and predicate calculus. 1) Each deals with a different type of elementary wff. Propositional calculus deals with an elementary wff which is in the form of a statement that cannot be decomposed. Predicate calculus, on the other hand, deals with and elementary wff which can be broken down into its component

symbols representing individual entities. Thereby allowing the system to carry out logical reasoning with these symbols in contexts other than that of the initial wff (Glorioso and Colon Osorio 1980).

2) Predicate calculus also has the ability to include variables in its reasoning process. A variable in predicate calculus is associated with one of two quantifiers which are "There exists" represented as $\exists$ and "For all" represented as $\forall$ (Raphael 1976). A variable associated with a $\exists$ symbol is said to have at least one value for which the wff it is contained in will be true. The $\forall$ symbol indicates that all values assigned to that variable will make the wff which contains it true.

Due to the means by which a problem may be represented predicate calculus is applicable to a wide range of problems. The atomic formula of predicate calculus consists of a variety of symbols which can be used in a number of ways to represent the integral components of a situation. These symbols are defined in the following manner by Nilsson (1980). Constant symbols, which are the simplest, are used to represent objects or entities. Function symbols are used to represent the operation or function served by elements of the domain. Variable symbols represent a variable or unidentified entity. Predicate symbols are used to signify some relationship existing between the elements of a domain. The various symbols just named are often grouped into two general categories: 1) functions—those symbols designating some existing relationship, and 2) arguments—those symbols representing elements of the domain.

The most common proof procedure for predicate calculus is an inference structure known as resolution. The resolution proof is a procedure which initially assumes the theorem to be false. Through a series of manipulations of these clauses, referred to as reductions, the resolution process attempts to prove the theorem by contradiction, i.e., if one cannot prove the assumed theorem true it must be false (Cohen and Feigenbaum 1982).

The initial step in the resolution process is to change all premises and the negated theorem into clause form. Clause form is a wff that is composed of literals with only "OR" connectives, i.e., a disjunction of literals (Henschen 1976). Literals are defined as an atomic formula or the negation of an atomic formula (Henschen 1976), e.g., x, -x, (4,-3). The clauses which have been constructed are then grouped together and searched until a clause containing some literal, x, and a clause containing its negation, -x, are located. These two parent clauses are then combined, with the common literal cancelling itself, creating a new clause without the common literal. A simple example of this process, call unit resolution, for creating a new clause is shown below in Figure 5.1. The new clause is then added to the initial group of parent clauses, replacing the parent clauses used in its formation. The process of selecting clauses and creating new ones is repeated until either no new clause is deduced or a contradiction is detected. If a contradiction is discovered, the theorem is proven. This basic outline is discussed by Cohen and Feigenbaum (1982) and by Raphael (1976) as a proof for propositional calculus.

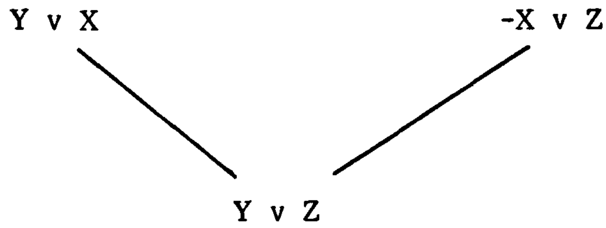


Figure 5.1 Example of Unit Resolution

The two most significant problems encountered when using resolution are changing premises to clause form and unification (Cohen and Feigenbaum 1982). The problems of changing premises to clause form occur when one is required to represent quantifiers of predicate logic in clause form. The problems of unification involve the determination of which arguments are comparable and what substitutions are acceptable, i.e., if x represents all integers, is 3 a viable substitute for x? Additionally, resolution by refutation, as it is often called, involves a considerable amount of time and effort when constructing a proof because of the amount of search and derivation involved in this procedure.

To help reduce this effort and time three, rather important methods are often used to improve the effectiveness of the search (Raphael 1976, Henschen 1976). These strategies attempt to help direct the selection of clauses to be resolved and the selection of literals within those clauses to be cancelled out. 1) The set-of-support strategy requires that the initial clause selected for each proof be a clause contained within the negated theorem to be proven. This strategy will limit the number of resolutions, by carrying out only

those resolutions most likely related to the theorem. 2) The unit preference strategy carries out all unit resolution prior to allowing non-unit resolution. Unit resolution is resolution of clauses which contain only one atomic formula, e.g., the previous example is unit resolution. This type of resolution can be done rapidly and easily, generating shorter clauses which assist in closing in on the desired refutation. 3) The linear format strategy attempts to direct the deductions in a straight line towards a refutation by continually using one of the derived clauses as a parent clause in each resolution. Although these three strategies are quite common, they are not the only ones. A number of other strategies have been developed to increase the efficiency of the resolution process.

The logics discussed thus far have all been first-order logics. It is important to realize that this is not the only type of logic which has received the attention of AI researchers. Some additional logics which have been studied are higher-order logics, modal logics, multi-valued logics and fuzzy logics (Raphael 1976). Their use is not yet common to many AI systems and they exist in a realm beyond the scope of this discussion.

5.2.5 Non-Monotonic Reasoning

Non-monotonic reasoning is a "common sense" form of reasoning which is able to make guided inferences while dealing with incomplete knowledge (Winograd 1980). This type of reasoning makes inferences that are believed to be the best at that instance but, at some point in the future, if they are proven false, can be invalidated

by the introduction of a new inference which updates the system. Non-monotonic logics serve as the judge to decide which inference or assumption is to be upheld and which is to be invalidated (Cohen and Feigenbaum 1982). Non-monotonic logics will be given the most attention in this discussion. However, first it is necessary to describe two important methods of non-monotonic reasoning.

The most common methods of non-monotonic reasoning include reasoning by default and reasoning by circumscription (Cohen and Feigenbaum 1982). Reasoning by default is a method which reasons by inferences or rules which are believed true until they are shown to be false, at which time a default occurs and a new inference is believed to be true. The method of reasoning by circumscription uses the same concept as reasoning by default but allows the system to reason only with those objects and relationships which are explicitly mentioned within the problem, i.e., no assumptions regarding information not mentioned within the problem can be made.

Non-monotonic logics were developed as a means of overcoming the problem encountered when using formal logics to reason with situations involving incomplete knowledge. To cope with incomplete knowledge, while developing the proof, non-monotonic logic provides that if in the course of the proof a previous assumption proves false it may be replaced by a valid statement or new assumption (McDermott and Doyle 1980). Non-monotonic logics gain some of this ability to handle incomplete knowledge by using formal logics with the addition of a term representing consistency. For an explanation of this method see McDermott and Doyle (1980).

Two additional concepts important to non-monotonic logic are inference rules and proof procedures. The inference rules of non-monotonic logic are quite similar in concept to those of monotonic logics in that they are the operators for manipulating the premises. However, the principle purpose of non-monotonic inference rules is to direct the process of selecting tentative beliefs (McDermott and Doyle 1980). Proofs seem to be somewhat elusive for non-monotonic systems. McDermott and Doyle (1980), state that the only adequate solutions known are the Truth Maintenance Systems (TMS), developed by Doyle, and its relatives.

The two main functions of the TMS are: 1) maintaining a current data base of generated proofs, and 2) the detection and elimination of inconsistencies within the database (McDermott and Doyle 1980). The TMS is used in the problem-solving process to collect bits and pieces of deductions which are made throughout the reasoning process. TMS then uses these deductions, or "pieces of proofs," to continually update the database (Doyle 1979). The second function of the TMS is accomplished by keeping track of "justifications" for each formula or belief in the data base. This gives the system the ability to realize why a belief was assumed correct and assists in the determination of which formulas need to be updated in the event of a changed belief (McDermott and Doyle 1980). A more complete description of the TMS may be found in Doyle (1979), McDermott and Doyle (1980) and in Barr and Feigenbaum (1982).

Although non-monotonic logic systems seem quite promising for many AI applications, a number of problems still exist (McDermott

and Doyle 1980). Since there is no known procedure to identify theorems, the major problem is to determine if a statement is indeed a theorem and, if so, whether it is provable. Additionally, the logic of the system is rather weak in that it lacks complete consistency. The proof procedure, TMS, also has an inherent problem due to this inconsistency which causes the procedure to loop forever in certain instances.

5.2.6. Means-Ends Analysis

Means-ends analysis is another type of informal problem solving method, developed by Newell, Shaw and Simon as the problem solving mechanism for the General Problem Solver (GPS). GPS will be specifically dealt with later. However, the technique of means-ends analysis will be briefly discussed here. The means-ends analysis is a difference-directed inference system. That is to say, the inferences are selected on the basis of the difference which exists between the current state and the desired goal state. This selection process is a type of search and has been discussed in greater detail in Chapter 4 Section 4.7.4.

In the means-ends analysis process the states are described as objects and relationships which represent situations to be manipulated. The operators are the alternatives available which, when implemented, manipulate the states to lessen the differences existing between the present state and the desired state. Therefore, the basic problem amounts to the mere selection of the proper sequence of operators, based on the measured differences which exist (Raphael

1976). The two key processes of means-ends analysis are: 1) the method used to determine the differences existing between the current state and the desired goal state, and 2) the method used to select the "best" or most effective operator for reducing the detected differences.

The means-ends analysis method of solving problems is applicable to a wide variety of problems because of the generality of the method, i.e., it is not restricted to only certain problems by a highly structured set of rules. The major advantage to this method is that the sequence of operators is neither limited to any prespecified order nor restricted to only one application per operator for each proof (Raphael 1976). This advantage results from selection of the most effective operator for reducing the current differences without regard to prior selections or situations.

One of the disadvantages to this method of problem solving is that the method is often inefficient when locating an appropriate operator because the whole set of possible operators is searched at each point an operator is to be selected. Another disadvantage is that the representation of the objects of a large problem may be limited by the memory space available. Lastly, the structure of the "task domain" is not explicitly defined but must be set up by the user when building the system, adding to the difficulty of setting up the initial system (Raphael 1976). (The task domain refers to those objects, operators and differences the system keys on to detect existing differences, i.e., a domain set up by the user to be used by the system to direct the processes.)

5.2.7 AND/OR Graphs

No discussion of problem solving methods would be complete without mentioning one of the most elementary problem solving techniques. AND/OR graphs are a means of solving problems based solely on the problem-reduction structure of a tree, consisting of AND nodes and OR nodes, and the search strategy. Although this technique was covered in Chapter 4, Section 4.7.4 as a search strategy, a brief review of the major concepts at this point is appropriate. The process of AND/OR problem solving is to reduce the initial problem to a number of sub-problems or simpler equivalent problems which can be easily solved (Nilsson 1980). The AND nodes of the graph represent a set of sub-problems which together can be used to represent their parent problem, while the OR nodes represent a number of problems which are equivalent to but hopefully simpler in nature than the parent problem (Winston 1977). The methods of generating AND/OR graphs, i.e., search strategies, can be any one of a number of strategies which were discussed. It is also important to recall that AND/OR graphs are a useful means of representing problems which can be solved by reduction. However, the efficiency of this method ultimately depends on the efficiency of the search method which is chosen.

Before leaving this discussion of deductive problem solving it is important to realize that there is a wide range of problem solvers other than the few discussed here. A group which is receiving much attention is often referred to as non-resolution methods. These methods involve a wide variety of techniques which apply heuristic

information and "user supplied" knowledge to increase the power of the problem solvers (Bledsoe 1977). Many early problem solvers were based on the strict syntactic formalisms of resolution which were "unnatural" methods. However, this group of non-resolution techniques is regarded by some as a more "human-like" method of solving problems. Among the techniques of non-resolution problem solving are means-ends analysis and AND/OR graphs (with heuristic searches). Additional methods can be found in Bledsoe (1977).

5.3 Induction

Induction is another method of reasoning by derivation in which one reasons from specific truths to general truths. Inductive reasoning is the inverse of deductive reasoning. Because of this property much of the research on inductive reasoning has dealt with inverting deductive proof systems and determining their resulting effectiveness. Inductive reasoning is also used quite extensively when a system implements learning by example (Cohen and Feigenbaum 1982). In this type of learning the system must be able to make inferences from specific examples and reason from them to general rules which will be used to guide further actions. This method of learning rules by induction has been tested by Michalski and Chilausky (1980). Additionally, a method by which inductive rule acquisition is accomplished is discussed by Quinlan (1979).

The process of inductive reasoning often implements some of the methods for deductive reasoning such as propositional and predicate calculus representations and proof procedures. However,

this type of inverted deductive reasoning would not be considered "pure" induction. The major problem encountered when using induction is completeness of the proof. Completeness refers to the method's ability to prove all theorems in and of itself. When using inductive reasoning procedures which have been created by inverting a deductive reasoning procedure, the proof is often too complete (Meltzer 1970). For additional information on inductive reasoning see Barr and Feigenbaum (1981) and Cohen and Feigenbaum (1982).

5.4 Recognition

Recognition as the name implies is the ability of a system to recognize a pattern of data rather than the path of steps necessary to discover or prove the data. Recognition deals with the detection and classification of patterns. Patterns are described by Glorioso and Colon Osorio (1980) as information structures or signals with no convenient mathematical description. A pattern recognition or classification system works with the "features" of a pattern which are structures of the picture being examined giving it identifiable characteristics. The ability of a system to discern between the various features of a pattern is referred to as the discrimination of the system (Glorioso and Colon Osorio 1980). At present, most AI pattern recognition systems can handle only fairly simple patterns.

There are two common methods used for pattern recognition: the template approach and the n-tuple approach (Uhr 1973). The template approach, often called the structural approach (Glorioso and Colon Osorio 1980), uses a data base of pattern representations

which serve as templates for the matching process. Recognition is then accomplished by attempting to find the "best" match of the current pattern to the templates stored in the data base. This method is obviously limited by quality of the match which can be achieved and by the number of templates which can be stored in the system's memory.

The n-tuple method, also known as the discriminant method (Glorioso and Colon Osorio 1980), represents each pattern by storing n matrices containing information about the various features of the pattern. The most common form of this method is the 1-tuple method in which one matrix is stored in the systems memory for each pattern (Uhr 1973). The values of the matrix are most often determined simply by placing a grid over the pattern to be represented. The probability of containing a portion of the pattern is recognized simply by transforming it into a matrix by some method, similar in nature to the one just described. The matrix is then evaluated against the matrices of the system's data base for a most probable classification or "best" match. While the 1-tuple approach uses only one matrix to represent one pattern the n-tuple approach uses n descriptive matrices to represent one pattern in an attempt to increase the accuracy of the representation.

Current applications of pattern recognition systems include the recognition of simple symbols and shapes and the classification of complex wave forms such as sonar and brain waves (Raphael 1976). The major disadvantage associated with both the template and the

n-tuple approach is their inability to effectively deal with patterns which have been shifted from the position in which they were initially modelled. Also, the number of patterns that can be identified by both methods is ultimately limited by the size of the representation for each pattern and the amount of memory space available (Glorioso and Colon Osorio 1980).

An interesting approach which helps to decrease the amount of memory space required for a large number of patterns is discussed by Glorioso and Colon Osorio (1980). The method is referred to as syntactic pattern recognition and seeks to represent only primitive pattern forms in the data base. Complex patterns are then described in terms of a number of these primitive patterns and a set of rules for constructing the complex pattern from the primitives. In this discussion of pattern recognition techniques only the most common methods were briefly described. For additional information see Uhr (1973) and Cohen and Feigenbaum (1982).

5.5 Meta-Level Reasoning

Thus far this discussion of problem solving has dealt with reasoning at the object-level. It is also necessary to mention briefly how reasoning is handled at the meta-level, i.e., reasoning about how to control effectively the large amounts of knowledge represented within a system. Meta-level reasoning is needed for systems with large data bases which encounter a situation referred to as saturation (Davis 1980). Saturation is a condition which exists when the amount of knowledge retrieved at a given instance by the system is so large

that exhaustive invocation would be impractical. Meta-level reasoning is essentially a method which is used to manage the knowledge of the system.

Davis (1980) gives a brief discussion of a number of methods which have been implemented to manage large amounts of data and a description of his own recently developed means of handling these types of knowledge management problems. The first method mentioned, procedure invocation, was used in the GPS by Newell, Shaw and Simon. In this method the order of invocation and priority of each inference rule is determined when the code is being made, i.e., the information which deals with reasoning control is "hardwired" into the system. The problem created by this method is that the system has no means of refining or changing the strategy of control when the system is being run. This causes a deterministic strategy for reasoning about control.

A second method which has been used to guide a system's invocation of rules was implemented by production systems. This method used a "conflict set" which contained all the rules which could potentially be activated at that instance. The system's interpreter was equipped with a means of prioritizing the rules, thereby selecting a firing order for the most applicable rules.

Additional methods include the use of theorem proving strategies, such as in QA3. This method attempted to prune the applicable clauses which could be resolved, thereby changing the structure of the initial inference engine. PLANNER implemented a "recommendation list" which contained recommendations for achieving

a goal, given a certain context. CONNIVER, on the other hand, used a "possibilities list" which listed all the statements which matched the current pattern. Additionally, NASL contained a "choice mechanism" which allowed the system to keep a list of rules which could be used to guide the selection process when a number of possible statements were applicable at one instance.

The method developed by Davis, as part of Teiresias, uses meta-rules for reasoning about control. Meta-rules are rules which specifically deal with how knowledge is to be manipulated. These rules are quite similar in principle to the operators which are used at the object-level. Davis gives three major advantages to this means of controlling the reasoning process. 1) The control information is separate from the object-level knowledge, alleviating the problems of accessing only the object-level knowledge without its associated control information. 2) The control information, since it is a separate group of information, can be directly examined and manipulated by the system. 3) The knowledge at both the object-level and the meta-level has the same form of representation. This allows the object-level and metal-level knowledge to be manipulated by the same inference engine, thereby reducing the overall size of the system. It is important to add, however, that a number of disadvantages to this method have been encountered (Davis 1980). These include, among others, difficulty in pursuing a single line of thought, and at present, an insufficient vocabulary for representing the concepts of meta-level knowledge. For additional information on meta-level reasoning see Davis (1980).

5.6 Summary

At present researchers are investigating a variety of methods for problem solving at both the object-level and the meta-level. Most problem solving techniques currently solve problems through the process of derivation or recognition. Derivation processes can be divided into deductive and inductive reasoning. Deduction is used to solve a wide variety of problem types and implements any one of a number of proof procedures depending on the problem type at hand. Induction, on the other hand is used most often to generate rules from examples as part of a learning process within the system or in combination with deductive reasoning processes. Recognition processes, which are still in rather elementary stages of development, are used to solve problems involving pattern identification or classification. Lastly, it is important to realize that the selection of any one of these problem-solving techniques is highly dependent on the type of problem to be solved, the amount and type of information available about the problem and the method of representation of the problem.

CHAPTER VI

ACCOMPLISHMENTS OF ARTIFICIAL INTELLIGENCE

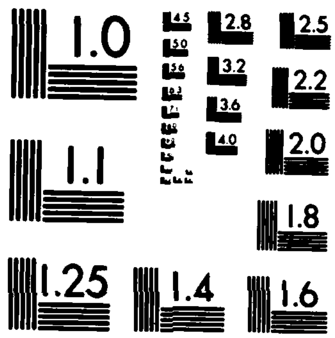
6.1 Introduction

In this section a number of AI programs which have been developed to exhibit some form of "intelligent" behavior will be briefly described. Although some of the programs to be mentioned are in the development stages, others were developed in the early days of AI, contributing greatly to the development of future programs through some significant feature. Additionally, it should be noted that some of the programs to be described fit into more than one of the sections even though they may be mentioned in only one.

The first section includes programs which are able to play games such as chess, backgammon and checkers and programs for solving general problems. Programs which are used to prove theorems, such as symbolic logic theorems, will be discussed next. These two sections deal with programs whose capabilities closely resemble those which one would expect from techniques discussed in previous sections; therefore, there will be very little discussion of their background. Each of the remaining four sections, however, will include a brief discussion of the area of AI with which they are concerned.

AD-A146 681 THE APPLICATION OF ARTIFICIAL INTELLIGENCE TO CONTRACT MANAGEMENT(U) CONSTRUCTION ENGINEERING RESEARCH LAB (ARMY) CHAMPAIGN IL T A KRUPPENBACHER AUG 84

UNCLASSIFIED CERL-TM-P-166 F/G 9/1 NL



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

Programs which deal with natural language processing will be discussed in the second section. These include programs having a limited ability to process and understand either written or spoken language. A few programs which exhibit some form of learning or planning ability are briefly explained. A few requirements of programming languages which have been used for AI systems and a few of the programming languages which have been used for AI systems are mentioned. The last discussion focuses on expert systems which exhibit the capabilities of expert consultants or act as "intelligent agents."

6.2 Game Playing and Problem Solving

Ever since the beginning of computers there have been programs which were able to play games effectively. Today there are AI programs which are capable of playing games such as chess, checkers, backgammon and a host of others at amazing levels of proficiency. These programs implement strategies which, in most instances, are similar in principle to those used by expert players but with blinding speed. The single most important technique in most game programs is the method used to search the myriad of possible moves contained within a game tree. The method used must not only be efficient, knowing which branches to expand and which to ignore, but must also be able to recognize a "best" move. To accomplish this, many techniques implement various heuristics, gained from expert players, to guide the search process.

Possibly the most popular game to be programmed into a computer is chess. A number of systems presently play at a human rating of Expert. The first program to achieve this level was CHESS 4.5, developed by Slate and Atkin in 1977 (Cohen and Feigenbaum 1981). This program used an alpha-beta search technique with a one step progressive deepening strategy and a preset limit on search time per move. The system also saved the results of previous iterations as a reference for later moves (Cohen and Feigenbaum 1981). This particular method is reported to have searched an average of 500,000 nodes per play and penetrated approximately six ranks into the tree. At present, however, Waltz (1982) states that the current world champion of chess playing systems is the Belle system, developed by Thompson and Condon of Bell Laboratories. The system operates on a special computer designed specifically for analyzing chess moves. Guided by a number of heuristic rules which help prune the game tree being searched, Belle analyzes approximately 29 million moves in the three minute time interval allowed per move.

Other games which have recieved much attention are backgammon, checkers, draw poker and go. Hans Berliner has developed a program known as Mighty Bee for playing backgammon. This program defeated the 1979 world backgammon champion (Waltz 1982). During the 1950s and early 1960s A. Samuel developed a program for playing checkers which learned the various positions which were evaluated during training and then recalled these evaluated positions while searching for the "best" move (Samuel 1963). The program implemented a mini-max look-ahead search with evaluation functions

developed from an average of twenty to thirty features of each position. Despite its ability to pick the best or second best move 64% of the time, it was still defeated by the world checker champion in 1965 (Barr and Feigenbaum 1982). For more information on these and other game-playing programs see Barr and Feigenbaum (1982), Waltz (1982) and Feigenbaum and Feldman (1963).

Problem-solving programs have been developed to solve a variety of problems such as getting from one situation to another. The programs mentioned here include General Problem Solver (GPS), developed by Newell, Shaw and Simon; STRIPS, developed by researchers at the Stanford Research Institute and AM developed by Lenat. GPS was developed as a system which would be able to solve any type of problem using difference-directed inference procedure guided by heuristics. Although the system worked quite well on a number of applications and contributed some new developments to AI work, it was unable to solve "any" type of problem as had been initially desired (Lenat 1978). For a review of the concepts of difference-directed inference procedures see Chapter 4 Section 4.7.4. The GPS was revised and applied to the STRIPS system by a team of researchers in 1969 (Raphael 1976). STRIPS is given an initial situation, a desired goal and a set of operators. The system implements a difference-directed strategy to determine the proper sequence of operators necessary to achieve the goal (Winston 1980, Barr and Feigenbaum 1982).

The AM program developed by Lenat is not a typical problem-solver. The system "performs scientific problem solving" (Lenat 1978,

pg. 249) by discovering mathematical concepts. AM begins with a number of "core concepts" and a few hundred heuristic rules which direct the process of determining specifics about the initial concepts. The system attempts to discover "interesting" concept segments to expand upon and from which eventually to refine a new mathematical concept (Lenat 1978). Although the program has found a number of concepts and made a number of conjectures about mathematical relations, the most significant weakness is that its heuristics are too weak to effectively guide the process to great depths of discovery (Waltz 1982, Lenat 1978). A derivative of the work on AM is EURISKO, by Lenat, which can be used in the discovery and development of new heuristics (Cohen and Feigenbaum 1981).

6.3 Theorem Proving

Although much of the use and implementation of theorem-proving has been referred to in previous sections of this paper, no program developed specifically for this purpose has been mentioned. Two programs will be discussed, one to prove symbolic logic theorems and the other to prove geometry theorems. Logic Theorist (LT) was developed by Newell, Shaw and Simon in 1956 to investigate the use of heuristics in problem solving (Cohen and Feigenbaum 1981). The problems LT was to solve where symbolic logic theorems for which the program was to develop formal proofs using propositional calculus (Lenat 1978).

The program was given a theorem to be proven and set of axioms as well as some heuristics. The process of developing a proof

was to search exhaustively the possible operators in a backward strategy guided only by a few heuristics. The operators were applied in a set order to generate new states which were tested against the given axioms. If a match occurred, a path had been located between the initial theorem and that axiom, thereby proving the theorem (Cohen and Feigenbaum 1981). This method of applying logical operators and exhaustively searching the resulting states was an inefficient and rather weak method of proving theorems. Since its creation, LT has been revised to increase both its effectiveness and efficiency by applying additional features to limit the search and to change the number of given theorems and axioms (Cohen and Feigenbaum 1982).

The theorem-proving program to prove geometry theorems was developed by Gelernter and Rochester in 1958. The primary focus of this sytem, as with LT, was to investigate the potential of using heuristics. The theorems which this system could prove were relatively simple ones; however, more complex theorems could be proven with the addition of rules and an increased amount of search time. For a more complete description of this system see Gelernter (1963).

6.4 Language Processing and Understanding

Before addressing systems which have been developed to deal with the problems of understanding written or spoken natural language it is necessary to give a brief background on natural language processing and understanding. A number of aspects of natural language

processing are currently receiving much attention by AI researchers. The two most important of these are machine understanding of natural language and machine translation. Possibly one of the most difficult tasks faced by AI researchers is to create an AI system which is able to understand "everyday" language, not requiring the use of specialized vocabulary. Most language in use today to converse with a computer is structurally simpler yet syntactically more complex than our normal conversational language (Barr and Feigenbaum 1981). An additional point of interest is the development of a system which will be able to understand spoken language. The aspect of spoken natural language understanding will not be discussed here. For additional information see Barr and Feibenbaum (1981).

Machine translation is a variation of the natural language processing research in which work is being done on the use of a computer to translate from one natural language to another, i.e., Russian to English. Although machine translation is still limited in its ability, it is presently being used in a number of systems where the process is aided by a human (Barr and Feigenbaum 1981). The most significant conclusion drawn from the work on machine translation is that, for an accurate translation to be made, the system must understand the subject rather than merely manipulate the words. This conclusion obviously points to the need for systems which are able to understand natural language. For additional information on machine translation see Barr and Feigenbaum (1981).

When dealing with natural language processing, most systems are concerned with grammar and parsing. Grammar refers to the

method of structuring and interpreting a sentence based on a set of syntactic rules. Grammar is important to the interpretation and construction of natural language phrases. A discussion of some of the most common types of grammars can be found in Barr and Feigenbaum (1981). Parsing is the technique used to assist the interpretation process by breaking a sentence down into its components which can be more easily understood. An example of parsing would be to break a sentence down into its component verbs, nouns and predicates. To accomplish this breakdown, the system will apply the grammatical rules and any additional knowledge available in its data base to determine the functions and relationships of each word in a sentence (Barr and Feigenbaum 1981).

One of the most widely known programs seeming to possess language understanding abilities is ELIZA, developed by Weizenbaum in 1966. ELIZA plays the role of a psychiatrist carrying out a therapy session with a patient, the user. The program does not actually process the language in an effort to comprehend what is being input but merely implements a form of pattern matching. The input is simply matched against set patterns and key words within the data base. When a match is detected, a set response is given with certain key words from the input inserted at designated points within the response.

There are a number of programs which actually process natural language in an effort to understand it. Two of these are LUNAR, developed by Woods in 1972, and SHRDLU, developed by Winograd in 1972. LUNAR is capable of answering questions given

in everyday language regarding geologic information about the samples gathered during the Apollo-11 mission. The program operates by translating the question into what is referred to as a query language through the use of a parsing technique and a semantic interpretation procedure. The translated version of the question is then used to select the correct answer from the data base. The process outlined here is a simplified version of that described by Barr and Feigenbaum (1981).

SHRDLU, developed by Winograd, possibly achieved some of the most significant progress in understanding natural language. SHRDLU is a program which manipulates a one-armed robot according to instructions given during a dialogue with the user. The system operates in its own environment of blocks, i.e., a set of toy blocks of various shapes, sizes and colors, therefore giving it a concise and limited number of concepts. The system is able to deal with concepts involving objects, interrelationships of the objects, actions and the various features of the objects. When referencing an object, the user implements a description of that object, i.e., color, shape, size and location relative to other blocks, rather than calling each object by a specific name (Winston 1977).

The process used by SHRDLU to understand what is being input begins by parsing the sentence. In this process the sentence is broken down into meaningful segments, which are then broken down into meaningful groups of words, such as nouns and verbs. For inputs larger than a single sentence the first step is to establish the relationships between sentences. The syntactic units generated by

parsing then activate a corresponding semantic program which translates this unit into the designated response, such as a reply or a series of commands for manipulation of the blocks. The process described here is quite simplified. A detailed explanation at this point is beyond the scope and intent of this presentation; however, more information on SHRDLU can be found in Winston (1977) and Barr and Feigenbaum (1981).

6.5 Learning and Planning

Learning and planning are important functions of human development which are being implemented by some AI systems which exhibit intelligent behavior. Learning can be defined as an acquisition process where new skills or concepts are acquired which can be used to improve the performance of the system. Planning is the process of selecting a goal and then determining a series of subgoals or steps which can be followed to achieve the desired goal. Before discussing specific programs which implement either learning or planning a few basic concepts and types of each process will be mentioned.

Learning is often classified as four separate types which are defined in the following manner (Cohen and Feigenbaum 1982). Rote learning is the simplest form of learning where the knowledge to be learned is supplied in a form which can be directly stored or "learned" by the system, i.e., no processing of the information is necessary. The checkers-playing program developed by A. Samuel, mentioned in Section 6.2, uses this type of learning when new positions are encountered (Samuel 1963).

Advice-taking is a second type of learning where the system is supplied general knowledge regarding a domain. The information must be processed by the system and related to what is already known, i.e., the system begins with some initial knowledge of a given domain then expands on it through this type of learning. TEIRESIAS, a system developed to assist in the learning of expert rules, uses this type of learning.

The third process is learning from examples, or an inductive learning process. In this type of learning the system is presented with a number of specific examples from which the system must glean general rules regarding the domain. This method of learning new rules has been applied in a number of systems including Meta-DENDRAL and an unnamed system developed by Michalski and Chilausky (1980), which has been mentioned in Chapter 5 Section 5.4.

The fourth type of learning is learning by analogy. In this method the system accesses another data base containing knowledge regarding a domain similar to its own from which analogies can be drawn. From these analogies additional information is acquired which can be used to improve the system's performance when encountering situations that the data source from which the analogies were obtained is well suited for. This type of learning has not yet received much attention from AI researchers (Cohen and Feigenbaum 1982).

Learning from examples is used by a number of AI systems, one of which is Meta-DENDRAL. Meta-DENDRAL was developed by Buchanan and Mitchell to generate rules which will describe how

a structural family of molecules will behave in a mass spectrometer (Cohen and Feigenbaum 1982). To begin, the system is given a group of known molecules belonging to a structural family and their associated mass spectrographs. The process begins by statistically analyzing the given data to hypothesize which bond breaks resulted in which peaks of the given spectrum. The system then searches for rules which are supported by the hypotheses. The search progresses inductively from general rules to more specific ones until a desired degree of specificity is achieved. The third step attempts to refine and modify the rules (Feigenbaum 1978). Meta-DENDRAL is reported to have developed rules in a number of tests which are similar in quality to those developed by human experts.

TEIRESIAS, also used for learning, learns by taking advice. The process involves the acquiring of information from experts regarding a certain domain and transforming them into rules which relate to the knowledge base of the system. The system which TEIRESIAS learns for is MYCIN, an expert system which will be discussed in Section 6.7. TEIRESIAS operates as an expert's assistant helping to review the results produced by the MYCIN program. When errors are discovered by the expert, he can request that TEIRESIAS reveal the line of reasoning followed to produce the error. The expert then locates the error and through TEIRESIAS corrects the incorrect rule or adds new ones by giving TEIRESIAS the proper advice. TEIRESIAS then makes these adjustments within the MYCIN program and reruns the previous situation to check if the new results agree with the expert. This method of teaching MYCIN new rules through

the acquisition abilities of TEIRESIAS has been quite successful. For additional information on this system see Davis (1979) and Barr and Feigenbaum (1982).

Planning is the process of developing the sequence of steps necessary to reach a desired goal. Cohen and Feigenbaum (1982) describe four types of planning which have been studied in AI. Hierarchical planning, perhaps the most common type, is a scheme in which the highest plans represent abstract or simplified overall plans while the lower level plans deal with specific details of attaining a certain goal. The second type, non-heirarchical planning, merely outlines the steps to be followed to achieve a certain goal, paying no attention to their detail. Script-based planning, a third means of planning, implements a number of "skeleton plans" stored within the system. These "skeleton plans" represent an abstract outline of a plan which will achieve a certain goal. When the system selects a goal, it searches for the appropriate "skeleton plan" and then begins filling in the steps with specific details thereby developing a specific plan. Opportunistic planning, the fourth type presents a more flexible method in which the plan is developed in piece-meal fashion by organizing the various problem-solving operators which are "suggested" by the system (Cohen and Feigenbaum 1982).

A program which implements a hierarchical planner is STRIPS, mentioned previously in Section 6.2. The planning operation of STRIPS begins with a given situation and a desired goal. The program also has a set of operators which can be implemented to achieve a goal. The process of achieving a desired goal begins by planning a sequence

of operators, through the use of a means-ends analysis search. Operators are then selected by the search process which will have the greatest effect on reducing the differences. Each reduction produces a new set of preconditions resulting in situations and operations of finer detail, eventually achieving the desired goal. At this time a course of action or plan has been constructed which can be followed to achieve a desired goal (Cohen and Feigenbaum 1982, Raphael 1976).

Two systems called MOLGEN have been developed to assist molecular geneticists plan laboratory experiments involving DNA. One system, developed by Stefik in 1980, uses hierarchical planning. The hierarchy was accomplished by planning on three separate levels, ranging from specific to abstract in nature. Each of these levels is used to plan something different: the bottom level plans the specific objects and operators of the experiment, the middle level plans how the experiment plan is to develop, while the top level plans what planning strategy is to be followed (Cohen and Feigenbaum 1982).

The other MOLGEN system, developed by Friedland in 1979, uses a script-based or skeletal planning method (Cohen and Feigenbaum 1982). In this method the system is supplied with a data base containing a number of abstract skeletal plans and additional knowledge which can be used to give detail to the plan. The process begins by inputting a problem. The data base is then searched for a plan whose end result is most similar to the goal. The next step is to fill in each detail requested by the skeletal plan based on specified

selection criteria. Although this method of planning has been quite successful on simple levels, MOLGEN (by Friedland) still encounters problems of limited knowledge for more difficult planning problems. Only a few learning and planning systems have been mentioned in a very simplified manner. For additional information on these and other systems see Cohen and Feigenbaum (1982).

6.6 Programming Language

Programming languages are the tool by which the techniques and applications discussed in this treatise of AI are constructed. The programming language serves two main purposes in AI systems: 1) the means by which programs carry out thought processes might be constructed, and 2) the method by which concepts and knowledge might be represented. When considering a programming language for an AI system, four important features should be compared: 1) the data structure or data types available, 2) the control structure which is implemented, 3) the means of pattern matching, and 4) the programming environment or set of support facilities for the language. A discussion of these features at this point would be inappropriate; however, a comparison of a number of programming languages relating to these four categories can be found in Barr and Feigenbaum (1982). The discussion to follow in this section is not intended to review in any detail or evaluate the programming languages available but to mention briefly a few of the features available to give an awareness of the work in the area of programming languages.

The languages used to construct AI systems are high-level languages which can be categorized into two major types, block

structured or dynamic. Block structured languages consider programs as groups of block structures which confine the accessibility of variables to the blocks containing them and limit the access which procedures have to each other (Barr and Feigenbaum 1982). Block structured languages also define the amount of space available for data prior to running the program. Dynamic or LISP-like languages, as they are referred to by Barr and Feigenbaum (1982), allow variables to be carried down through the program, not confining them to certain areas. Additionally, procedures are allowed to call up any other procedures within the program and the allocation of data space is dynamic, i.e., memory is permitted to change size as required (Barr and Feigenbaum 1982). The LISP-like languages described above are the most common languages in AI today. The discussion to follow briefly mentions some of the important characteristics of LISP and a number of the more significant language systems, some of which are based on LISP.

LISP, developed by McCarthy in 1958, is a programming language which uses LISt Processing. LISP, the second oldest programming language after FORTRAN, is the most widespread language in use in the field of AI today. The basic data structure of LISP is the list which is composed of symbols referred to as atoms. A variety of list types exist, such as a property list which is used to represent relationships among symbols. An additional characteristic of LISP is that it is a recursive language, i.e., the definition of a function may contain the function as a part of it.

Programs which are written in LISP also have a number of important features which are mentioned in Barr and Feigenbaum (1982). The structure of a program in LISP is a list, similar in structure to data lists. This feature allows programs to manipulate other programs as well as themselves by the same means used to manipulate data. Another feature, the simple syntax of the language, allows programs which can be constructed to debug or write other functions and programs. A discussion of this process, often referred to as automatic programming, can be found in Barr and Feigenbaum (1982). Additionally, programs can be constructed in an incremental fashion, thereby facilitating the construction and debugging of a function prior to constructing the next function. This feature allows the construction of more complex functions from the combination of these smaller incremental functions. For a more complete description of LISP see Winston (1977), Maurer (1972) and Winston and Horn (1981).

LISP and its dialects have been the basis of many AI programs. Some of the more common dialects of LISP include INTER-LISP, MACLISP, QLISP, FRANZ LISP and UTLISP. The purpose of these dialects is to create large LISP systems with special functions and procedures to simplify implementation and provide a larger set of support facilities. The following languages which will be briefly mentioned were designed with a similar purpose in mind, i.e., each is meant to provide a simpler implementation for certain types of problems.

Two languages classified as deduction/theorem proving languages, PLANNER and CONNIVER, were developed in the early 1970s

contributed to the development of languages in use today (Fahlman and Steele 1982). PLANNER, developed by Hewett, constructed programs as a collection of statements called theorems. These theorems were used to supply the system with the necessary control information to achieve a goal. The PLANNER representation and control scheme has been described briefly in Chapter 3 Section 3.4.1. CONNIVER was quite similar to PLANNER in may aspects. However, it did not implement an automatic backtracking procedure when a failure was encountered during search and used "methods" instead of theorems (Barr and Feigenbaum 1982).

Specialized programming languages, such as SAIL and FUZZY, can often aid in the development of programs for a particular domain. SAIL was developed in 1969 by Stanford AI Laboratory for AI systems dealing with vision and speech understanding. SAIL is a block structure language which is based on ALGOL and is similar to many modern compiler based languages. The impetus for its development was the need by vision and speech understanding systems for a language which could not only manipulate symbols but also do arithmetic rapidly (Barr and Feigenbaum 1982). FUZZY was developed by LeFaivre in 1976 as a language which could be used to deal with information which is "fuzzy" or vague. The language is LISP-based, using a number of additional functions to allow the explicit or implicit representation and manipulation of vague knowledge (LeFaivre 1976).

The following languages to be discussed have been designed for applications dealing with the representation and manipulation of

legal principles. The London School of Economics is developing a language called LEGOL as part of a project by the same name. LEGOL, the language, is designed to express legal principles contained within legislation based upon the relationships existing within the legislation. The language is an important component of the LEGOL project which is a legal consultation system. The system has been successfully applied to a number of routine legislative problems and is currently being expanded to handle more complex legislative problems (Cook, et al. 1981, Niblett 1981).

ROSIE (Rule Oriented System for Implementing Expertise) is a language, recently developed by the Rand Corporation for the development of expert systems. Since the ROSIE system is written in INTERLISP, it is more appropriately referred to as a programming environment or a tool for building expert systems, i.e., most of the essential functions needed by the user to set up an expert system have been coded. The ROSIE language allows the user to express concepts in a vocabulary quite similar to English while providing a program which is quite readable and well suited to legal applications (Fain, et al. 1982). The Legal Decision-making System (LDS) is an important application of ROSIE which has been developed to model the legal decision-making process followed during the settlement of product liability claims (Waterman and Peterson 1981).

The languages mentioned briefly in this section represent only a few which have been implemented in AI work. The major purpose of any of these languages is to provide a means by which

large and complex amounts of knowledge can be organized, accessed and manipulated. Other important factors include the structure used to organize the knowledge, the means of adding new knowledge and deleting old knowledge and the limits imposed on the storage and retrieval of knowledge (Fahlman and Steele 1982). The research and development of new and revised programming languages for AI applications continues today, attempting not only to improve the existing languages but also to provide a more natural environment in which man and machine might interact.

6.7 Expert Systems

Expert systems, having their beginnings in the mid 1960s, are a new class of AI systems currently receiving much attention. An expert system is a computer program which is able to act as an intelligent agent or expert consultant. The system gains its abilities from two major components: 1) a knowledge base, which is a collection of facts and heuristics regarding a specific domain or area of expertise; and 2) an inference engine or mechanism which directs the manipulation of the knowledge base according to a set of rules for applying the knowledge (Quinlan 1980).

Much like all problem-solving mechanisms expert systems are best suited to certain problem domains. The characteristics of a problem suited for an expert system are: 1) it should be difficult for humans to solve, i.e., requiring someone with expertise in the domain, 2) it should contain a large number of possibilities to be

explored, and 3) it should be important that no possibility be overlooked or left unexplored (Barr and Feigenbaum 1982). Expert systems which have been developed span a variety of disciplines ranging from medicine to chemistry, from law to geology. In the following paragraphs only a few of these systems will be briefly mentioned. For additional information on expert systems see Barr and Feigenbaum (1982), Michie (1979) and Hayes-Roth, Waterman and Lenat (1983).

The MYCIN system, developed in the 1970s by Shortliffe, is quite possibly the best known and best engineered expert system in use today. MYCIN is used to diagnose bacterial infections of the blood and to recommend the appropriate medication. The MYCIN system acts as a consultant for a physician-user who answers a series of questions put forth by the system to help determine the diagnosis and recommendation. Two advantages of this system are: 1) its ability to explain the line of reasoning followed when asked to do so by showing the order of rules which were followed, and 2) its use of the specialized language of the medical profession thus overcoming many of the problems encountered when using common English (Feigenbaum 1978).

The MYCIN system has been very successful in its domain. A panel of experts judged the system to perform at their level in ninety percent of the cases examined (Feigenbaum 1978). A more complete explanation of the processes used by the system can be found in Davis, Buchanan and Shortliffe (1977). Two systems which have developed as a result of the work on MYCIN are TEIRESIAS and EMYCIN. TEIRESIAS, which was described in Section 6.5, is

used to acquire additional knowledge for the MYCIN system through a process of interactive learning sessions with experts. EMYCIN, on the other hand, is a skeleton version of MYCIN, i.e., MYCIN, stripped of all domain specific information, can be fitted with new rules to become the expert of another problem domain (Quinlan 1980).

PROSPECTOR, an expert system developed by Duda and Gaschnig in 1978, is an intelligent agent for the domain of mineral exploration and resource evaluation. PROSPECTOR is a rule-based system using a mixed initiative control strategy, i.e., the system uses backward chaining but the user is able to interrupt the system to change goals or offer information (Barr and Feigenbaum 1982). The system is able to indicate how favorable a geologic district or specific exploration site within a district is for a given type of ore and to evaluate a specific drilling location (Duda and Gaschnig 1981). The system was developed by transferring the expertise of geologists into rules which could be structured within the knowledge base.

Although PROSPECTOR had been tested on a number of developed exploration sites, its first actual test ended in September 1982. The PROSPECTOR system was given all the necessary information to evaluate the prospects of discovering porphyry molybdenum on Mount Tolman in the State of Washington. When PROSPECTOR then identified various areas and rated their favorability, these sites were then drilled to check the system's accuracy. The drilling found that the PROSPECTOR system had correctly identified and evaluated each of the sites. The results of this test can be found in Campbell, et al. (1982).

MYCIN and PROSPECTOR are only two of a number of expert systems which have been developed or are being developed. A few of these systems are listed here. Caduceus, in progress, is being developed as a consultant for doctors of internal medicine through the combined efforts of researchers at the University of Pittsburgh and at Stanford University in California. R1 is an expert system, developed by J. McDermott, which has been in use at the Digital Equipment Corporation since 1980 designing VAX computer systems. PUFF is a consultant for physicians dealing with problems of pulmonary functions. Others include MOLGEN for molecular genetics, DENDRAL for chemical structures and CRYSALIS which deals with protein structures. For more information on these and other expert systems see Feigenbaum (1978), Edelson (1982), Duda and Gaschnig (1981), Barr and Feigenbaum (1982) and Hayes-Roth, Waterman and Lenat (1983).

Although expert systems have received much attention in AI research laboratories, their use in the outside world has been quite limited. A number of problems still exist during the development stages of these systems, the most important of which are those encountered during the knowledge acquisition stage (Duda and Gaschnig 1981). This stage represents the phase during which the knowledge engineer works with experts of the domain determining the necessary rules and heuristics to guide each decision. The two major problems during this phase are the inability of the expert to determine precisely how he or she comes to certain conclusions and the time necessary to gather all the essential rules. A report by Edelson (1982) states

that for systems with more than fifty rules it takes approximately one hour for a knowledge engineer to get one finished rule from an expert. Additionally, he states that to build a complete system, containing 500 rules would take between five and twenty man years to complete. Although these types of problems cause expert systems to be quite expensive, much work is presently being done to develop new tools and techniques to assist the process of building expert systems. The developments which have been achieved from this type of work give much promise to widespread use of expert systems in the future.

CHAPTER VII

THE DIFFERING SITE CONDITION CLAIM

7.1 Introduction

At this time in our discussion it is appropriate to consider the problem domain to which the techniques of artificial intelligence will be applied. For this application of AI the construction claims aspect of contract management has been selected. However, since a variety of construction claim types exist, only one, the differing site conditions claim, will be the focus of this application of AI. The intent of this chapter is to define and explain the selected problem domain. The following topics will be addressed: what is meant by a differing site conditions claim, the various aspects of the differing site conditions clause which are important during the analysis process, the rationale behind selection of the differing site conditions claim for this application and the viewpoint from which the claim is approached.

Before proceeding with a discussion of the differing site conditions (DSC) claim it is essential to understand what a construction claim is. A construction claim is defined as "a request for additional compensation for damages or expenses incurred during the performance

of a construction contract."[1] Claims can arise during any phase of construction resulting from any number of situations. Although a variety of claim types exist only the DSC claim will be discussed here.

7.2 Overview of Differing Site Conditions

The DSC claim results from a clause within the contract which grants the contractor a remedy for additional expenses resulting from latent physical conditions at the site which differ materially from those indicated by the contract documents or from those conditions which could ordinarily be expected. The clause used for this application is the "Differing Site Conditions" clause which is part of the Code of Federal Regulations (CFR 7-602.4) and is clause 4 in the General Services Administration (G.S.A.) Standard Form 23-A contract. The clause, which is often regarded as the model DSC clause, reads as follows:

> (a) The Contractor shall promptly, and before such conditions are disturbed, notify the Contracting Officer in writing of: (1) subsurface or latent physical conditions at the site differing materially from those indicated in the contract, or (2)unknown physical conditions at the site, of an unusual nature, differing materially from those ordinarily encountered and generally recognized as inhering in work of the character provided for in this contract. The Contracting Officer shall promptly investigate the conditions, and if he finds that such conditions do materially so differ and cause an increase or decrease in the Contractor's cost of, or the time required for, performance of

---

[1]Vanden Bosche, A Construction Claims Dictionary, s.v. "Construction Claim" (1981).

> any part of the work under this contract, whether or not changed as a result of such conditions, an equitable adjustment shall be made and the contract modified in writing accordingly.
>
> (b) No claim of the Contractor under this clause shall be allowed unless the Contractor has given the notice required in (a) above, provided however, the time prescribed therefore may be extended by the Government.
>
> (c) No claim of the Contractor for an equitable adjustment hereunder shall be allowed if asserted after final payment under this contract.

Before delving into the major areas of importance in the DSC clause a few brief comments are in order. One of the major purposes of the clause is to limit the risk in bidding and eliminate large contingencies which attempt to account for possible costs which may be incurred as a result of encountering a differing site condition (Currie, et al., 1971). Additionally, without a clause of this type within the contract the contractor has no implied legal right to collect excess costs or damages due to a differing site condition.[1] The clause has been judged to be a fair and equitable remedy to all parties involved in the contract. Furthermore, the DSC clause is not intended to be an exculpatory clause,[2] i.e., a clause intended to shift the liability for physical conditions encountered.

---

[1] United States v. Spearin, 248 U.S. 132 (1918).

[2] James Julian, Inc. v. President and Commissioners of Town of Elkton, 341 F. 2nd 205 (1965).

7.2.1 Specifics of the DSC Clause

Before discussing the rationale for selecting this particular type of claim as the problem domain it is necessary to explain the important aspects and legal implications of the DSC clause. One of the most important aspects of this clause is that it addresses two types of conditions, often referred to as Type I and Type II conditions. Other important aspects addressed by the clause which will be touched on are: the nature of the condition for which the claim is being asserted, the site investigation requirements, the obligation of the contractor to give notice of the discovered conditions and the time at which this notice must be given.

As with any legal document it is important to realize that the full meaning of the text cannot be expounded upon with only a few words. Therefore, the intent of the following explanation is to give the reader a working knowledge of the major concepts which are of importance to the legal analysis of the DSC claim. For a more comprehensive treatise of the clause being discussed see Currie, et al., (1971).

The Type I condition is defined as "subsurface or latent physical conditions at the site differing materially from those indicated in this contract."[1] The most important characteristic of a Type I condition is that the conditions discovered at the site must differ materially from those conditions indicated by the contract. The key word is "indicated." For a claim to be valid under a Type

---

[1]G.S.A. Standard Form 23-A, clause 4, paragraph a.

I condition the contract must either explicitly or implicitly indicate some existing conditions at the site. If the contract is silent regarding physical conditions at the site, a Type I condition cannot exist. To obtain an equitable adjustment for costs incurred due to a Type I condition it is only necessary for the contractor to prove that the conditions discovered at the site were materially different from those indicated by the contract.

The Type II condition is defined as "unknown physical conditions at the site, of an unusual nature, differing materially from those ordinarily enountered and generally recognized as inhering in work of the character provided for in this contract."[1] Therefore for a Type II condition to exist the conditions at the site must be previously unknown conditions which are unanticipated or of an unusual nature. A Type II condition can exist even if the contract is silent regarding physical conditions at the site.

Proving the existence of a Type II condition is more difficult, however, than proving a Type I condition. To prove the existence of a Type II condition the contractor must first establish what the expected or usual conditions at the site are. Additionally, the contractor must prove that he or she did not have knowledge of the existence of the unusual condition prior to the bidding and that the condition could not have been anticipated by the reasonable contractor. One other point which needs to be mentioned is that the unusual condition need not be mutually unknown, i.e., neither the contractor

---

[1] G.S.A. Standard Form 23-A, clause 4, paragraph a.

nor the government knew of the unusual condition. The contractor can still recover for a Type II condition if the government had prior knowledge of the unusual condition even if the contractor had no reason to be aware of the condition.

Type I and Type II condition claims contain elements which are common to both. These include such items as the limitations of the conditions covered, site investigations and notice of the claim. The DSC clause, as used by this thesis, is not limited only to subsurface conditions[1] but also allows for recovery when aboveground conditions are encountered which differ materially from those conditions indicated or from those conditions which could reasonably be expected. However, because the recovery under this clause is limited to physical conditions at the site, recovery through the DSC clause for losses or damages resulting from economic, political or governmental conditions is not allowed.

Site investigations also play an important role in determining whether or not a contractor has the right to collect through the DSC claim. If a reasonable site investigation by a layman contractor would have revealed the actual or unusual conditions at the site, the contractor will not be allowed any form of adjustment. However, if a reasonable site investigation was not possible prior to bidding, the contractor can recover for losses or damages resulting from the differing site conditions. Additionally, if a reasonable site investigation was possible and the contractor failed to make it, the contractor

---

[1] Foster Constr. C.A. and Williams Bros. Co. v. United States, 435 F. 2nd. 873 (Ct. Cl. 1970).

essentially accepts the risk of encountering any differing site conditions (unless, of course, such an investigation would not have revealed the actual or unusual conditions at the site).

Finally, giving notice of the occurrence of a differing site condition and the timeliness of this notice are important factors for determining whether a contractor can recover for the encountered condition. Although the contract specifies that a notice shall be given "promptly and before such conditions are disturbed . . . in writing,"[1] meeting this condition is not always necessary for a contractor to recover under the DSC clause. (It is, however, the best policy to follow.) One additional point regarding the timeliness of the notice is that if final payment has been made prior to the assertion of the DSC claim no recovery is allowed.[2] Three conditions exist where the contractor may recover even though he or she may not have fully complied with the notice requirement: 1) if oral notice was given, 2) if constructive notice was received, and 3) if the government was not prejudiced because of the non-compliance.

The first two conditions, listed above, which can occur whereby the contractor may still recover refer to additional types of notice. They are: 1) the contractor has given oral notice of the condition to the contracting officer or his representative, or 2) the contracting officer or his representative has received constructive notice of the condition. Oral notice is self-explanatory; however, constructive notice is not as obvious. Constructive notice refers to the situation where

[1] G.S.A. Standard Form 23-A, clause 4, paragraph a & b.

[2] G.S.A. Standard Form 23-A, clause 4, paragraph c.

a contracting officer or his representative should have been able to realize, from his own visits and site investigations, that a differing site condition had been encountered by the contractor.

Prejudice is the second factor which enters into the decision of whether or not to allow the contractor to recover for a differing site condition when he failed to give prompt written notice of the condition. Prejudice occurs when the government is unable to verify the condition or when the government can show that a cheaper or more efficient resolution of the condition could have occurred if proper notice had been given. If the government has been prejudiced, the contractor will not be allowed recovery.

7.3 Rationale for Selection of the Differing Site Condition Claim

The DSC claim was selected from among the various types of construction claims for a variety of reasons. The most significant reason for selecting this claim is that its subsequent litigation is intended to show that a contractual right to recovery exists, whereas the litigation resulting from many of the other types of construction claims is intended to show that some form of breach has occurred. Since it is easier to prove the existence of a contractual right than to prove the occurrence of a breach of contract, the litigation necessary for a DSC claim will be less complex than that encountered in many of the possible construction claims. (Figures A.1 through A.23 in Appendix A give an outline of the analysis processes used to decide a DSC claim.)

Another reason for selecting the DSC claim is that the corresponding clause found in the G.S.A. Standard Form 23-A contract is a concise and well-written clause. In fact, this clause is often referred to as the "model clause" and has been used quite frequently as a template for similar clauses in many other contracts (Simon 1979). Widespread use of the clause as a model further points to its legal credability in that it has been the focus of a number of court cases which have given it a relatively well-defined and well-tested meaning in the courts of law and in the various Boards of Contract Appeals. (For a partial listing of cases which have dealt with differing site conditions claim see Appendix D.) An additional reason for selecting this clause is that it is a separate clause within the contract and not referenced by or linked to any other clauses. This separation allows it to be treated as a self-contained legal concept, thereby, reducing the complexity of the analysis.

The DSC claim has also been selected because in most cases this claim is not inter-related with other types of claims. This is best illustrated by the decision trees in Figures A.1 through A.23 (in Appendix A) which represent the analysis process. Notice that: 1) a minimal amount of information is required from other areas within the contract, and 2) the only inter-relationships with other claims occur after is has been determined that no recovery is possible through the DSC clause.

Lastly, in the ideal case the DSC claim will result in very few decisions which must be based solely on professional judgement

if accurate records of the project have been kept. This aspect of the claim makes it more suitable for computerized legal analysis since most decisions will be "cut and dried," if the necessary facts are present. However, based on the research which has gone into the development of the decision tree for this claim, it has been found that the ideal case is rare. In fact, in a number of instances different courts have reached different decisions for cases which seemed to display similar circumstances.

7.4 A Consultant on Differing Site Conditions

Having selected the DSC claim for this application the remaining problem was to construct an AI system capable of implementing legal expertise to accurately predict the outcome of the claim based on the available information. This AI system is capable of assuming the role of a legal consultant to a field engineer or a lawyer. Before leaving the discussion of the differing site condition a few comments are necessary to explain how the system developed would function as a legal consultant and from what perspective the system analyzes the DSC claim.

To act as a legal consultant the system must not only be able to predict the outcome of the claim but must also have the ability to give advice. The system developed by this thesis can do just that. The basic assumption is that the DSC claim has been asserted at the jobsite. It is also assumed that the field engineer has sufficient technical knowledge of events and circumstances at the site. Therefore, the field engineer or field personnel requires

only legal advice as to what important facts are essential for justification or denial of the adjustment. This system gives the field engineer the ability to hold a meeting with an artificially intelligent legal expert to obtain the necessary advice, thereby eliminating the need for an additional person at the site to act as a legal advisor. During this meeting there is a question and answer period in which the artificially intelligent legal expert attempts to gather the information pertinent to the claim. The answers to these questions are based on the field engineer's knowledge of events and daily records.

Based on the information received the system analyzes the claim and attempts to fill in missing bits of information by instructing the field engineer as to what additional information must be gathered. If this information is unavailable, the system is able to assume a value and proceed with the analysis. At the end of the analysis the system gives the field engineer the expected outcome of the claim accompanied by a list of assumptions which were necessary.

Since the system functions as a legal consultant, the field engineer can at this point begin to substitute answers for the assumed values to test hypothetical or suspected situations which might occur. This aspect of the system's capabilities is also attractive to a practicing lawyer who may be seeking information regarding the effects a different set of facts might have on the outcome of the claim.

One last point which must be stressed is that the system views the claim from the government's or owner's perspective. This means that the questions which are asked are directed toward

the contracting officer or his representative at the job who has knowledge of the contractor's actions. The claim is viewed from this perspective because the selected clause is taken from the Code of Federal Regulations and the G.S.A. Standard Form 23-A contract.

7.5 Summary

This thesis attempts to apply successfully the techniques of AI to some aspect of contract management by constructing an AI system which can accurately predict the outcome of cases involving the DSC claim. This chapter has dealt specifically with the definition of the G.S.A. Standard Form 23-A DSC clause. The discussion of the clause explored the legal implications of Type I and Type II conditions, the importance of site investigations and the significance of the notice requirement. The reasons for selecting the DSC claim and the corresponding clause from the Form 23-A contract were also discussed. Additionally, a short discussion was included describing how the system which has been developed might function as an expert consultant to field personnel or a lawyer.

CHAPTER VIII

ARTIFICIALLY INTELLIGENT LEGAL ANALYSIS SYSTEMS

8.1 Introduction

This chapter presents the remaining background information essential for understanding the basis for the DSC legal analysis system. The discussion deals with previous work on computerized legal analysis systems and is divided into three major sections. The first section discusses the compatability which has been found to exist between law and computer programs. In the second section, some of the previous work which has been done on legal analysis systems will be discussed by commenting on current systems which have been or are being developed. In the third section, some speculation is made about the features which would be part of the "ideal" legal analysis system.

Some background information is essential to the discussion. The application of computers to law is possible in two major areas: documentation and consultation. A documentation system should be able to locate and retrieve legislation or case law concerned with . given concept quickly and comprehensively. Documentation systems are beginning to receive widespread use today. A consultation system,

also referred to as a deontic system, is capable of giving or obtaining legal advice, thereby acting as an intelligent agent concerning an area of law. Although these systems have not been developed to the level of ability required for widespread use, they are presently the focus of much research.

8.2 Compatability of Law and Computers

Although the compatability of law and computers may be evident, it is necessary to emphasize the existing similarities. The most important similarity is that both laws and computer programs require methodical processes of analysis (Fiedler 1980). Other similarities which make laws and computer programs compatible are cited by Niblett (1980) and include the following:

1. Laws and computer programs are composed of rules which are applied to facts and data, respectively.
2. Laws and computer programs require much precision when being created with respect to semantics and syntax.
3. The meaning of laws and computer programs lies in the result which they produce.
4. Laws and computer programs are of a descriptive and operative nature, i.e., both describe how a task is to be performed and both are functional parts of a system.
5. Bugs in laws and computer programs do not become obvious until they have been put into operation.
6. Laws and computer programs can contain ambiguities as a result of multiple interpretations.

7. Laws and computer programs can be expressed in propositional logic.

The existing compatibility of laws and computer programs has been known for over two decades. As a result researchers have been interested in the development of computerized legal analysis systems since the early 1960s. A number of benefits which exist for the development of such a legal analysis system include a more complete and methodical study of the legal analysis process resulting from the development of models of the lawyer's thought processes (Buchanan and Headrick, 1970). Modelling a lawyer's thought process would in turn cause lawyers to gain greater insight into their reasoning process. Another more distant benefit is the possible effect of a consultation system, which could accurately predict the outcome of cases, on the congestion of present day court systems (Stone 1964). A system which could reliably predict cases even for a small number of specific areas of law could reduce the backlog by allowing out-of-court settlements based on the system's analysis of the case.

An additional benefit, which would be more immediate to the legal analysis system developed by this thesis, is that a field office would have the expertise of a legal consultant without requiring a lawyer to be present at the site. By having the necessary legal expertise available through a computer the field engineer need only supply the technical knowledge as requested by the system. The system could then analyze the available information in light of its legal knowledge and supply the necessary legal advice.

The many similarities between laws and computer programs and the possible benefits of a computerized legal analysis system support the great potential that exists for the application of computers to legal analysis. Perhaps the major obstacle to the widespread development of computerized legal analysis systems lies in the language they use. Laws are expressed in natural language using all the richness of meaning and interpretation while computer programs are expressed in an "artificial language" which loses much of the meaning of a law through translation, often as a result of a limited vocabulary. The language difference is often regarded as the major difference between laws and computer programs (Fiedler 1980).

Other problems which have been encountered when attempting to construct a consultation system include: finding the appropriate rules to encode and finding the proper representation of the problem (Buchanan and Headrick 1980). Finding the appropriate rules to encode refers to the cumbersome task of analyzing a lawyer's thought process to discover, refine and encode the rules which are part of the lawyer's analysis of the case. Finding the proper representation of the problem involves selecting the proper means for representing the rules and facts which make up the problem.

8.3 Current Legal Analysis Systems

Currently, a number of legal analysis systems are being developed for a variety of areas of law. The legal analysis systems which will be dicussed in this section can be divided into two major categories: those systems using mathematical models and those using

AI techniques. A third group of systems, which will be briefly mentioned, has been developed with other aspects of law as its primary concern, such as documentation or the development of a computer language specifically for the expression of legal concepts. This group has attempted to accomplish a very shallow form of legal analysis.

A point which should be noted at this time is that legal analysis systems can be of two types (Buchanan and Headrick 1970). The most prevalent type is a system capable of analyzing an event that has already occurred where all the facts are known. The user of this type of system is interested in what his or her rights, remedies and risks are. This type of system analyzes the case by linking rules with the important facts to construct the most likely outcome.

The second type of legal analysis system is more of an advisory or planning system where the event has not yet occurred, or is developing, and the user is looking for guidance. This type of system assumes that the user has some control over the future events and will be able to use the system's advice to achieve the most desirable result. A system of this type would analyze the possible sets of events which could be forced to occur using some type of risk assessment to determine the best result with the least risk involved.

The mathematical models referred to in this section are primarily intended for predicting the outcome of a case based on a mathematical model of previous decisions of the judge or court hearing the case. This technique is based on the concept of stare decisis, i.e., "let the prior decision stand," in cases having the same

facts. A wide variety of mathematical models have been used to attempt to predict a judge's decision. Some of these techniques include: regression analysis, probit analysis, scalogram analysis, Bayes theorem and linear programming (Lawlor 1981).

The major problem with the mathematical method of predicting cases is that very few cases have the same facts and a sufficient number of cases are required to construct a reliable model. Due to this, most mathematical models in use must relate the presence and absence of facts in the case with those present in the previous cases of the same nature (Lawlor 1980). Another significant problem with this technique is that mathematical models cannot be universally applied but must be constructed specifically for each judge or court because many personal factors, such as experience and personal convictions, enter into the decision process (Lawlor 1981). Other problems which have become apparent are that of developing a reliable mathematical model, the cross validation of this model on the data base of cases and the testing of new cases not already present in the data base (Lawlor 1981).

An example of a system being developed which uses a mathematical model is SARA. SARA is being developed to predict cases involving discretionary decisions which can be based on discretionary norms. To analyze a case the lawyer assists SARA in the creation of a set of categories of facts which he deems to be pertinent. The lawyer also sets the specificity of the various facts and substantiates each fact with a previous case, if desired. This format of

facts then represents a model of decisions which the lawyer believes enter into the case at hand. The lawyer then evaluates each category of facts as positive or negative depending on whether or not the present case being evaluated contains such a fact. SARA then assigns weights to each of the facts which have been evaluated using any one of a variety of prespecified methods, such as a correlation method, and outputs the results. These results then indicate the importance that each of the different facts represents when considering the discretionary norm, thereby indicating whether the lawyer's argument is valid (Bing 1980).

The second category of legal analysis systems is made up of those which implement AI techniques. Many of these techniques were discussed in earlier chapters, therefore, they will not be dealt with in detail here. The intent of systems based on AI techniques is to give the system knowledge of the legal aspects and/or analysis process pertaining to a certain area of law. This knowledge can then be used by the system to answer questions, give advice or analyze a given set of facts in the particular area of law.

Initial speculation about legal consultation systems based on the principles of AI suggests that the system should attempt to follow the same thought process as a lawyer. This process has been broken down into four steps by Buchanan and Headrick (1970):

1) Examining the case to find the important facts
2) Selecting the most applicable or best rules
3) Finding the most effective linkage of facts and rules to support the argument

4) Finding analogies, i.e., other cases to support the argument

Although no programs developed to date adhere strictly to this procedure, a number of them are quite close. Two legal analysis systems that are based on AI techniques will be discussed to help explain a few of the methods which have been developed and to highlight the resulting capabilities and problems encountered through the use of these systems.

The TAXMAN Project is an attempt to apply the techniques of AI to the legal analysis of corporate tax law for corporate reorganizations. This project has resulted in the development of two systems, TAXMAN I and TAXMAN II. The TAXMAN I system is written in the micro-PLANNER language, while the TAXMAN II system is written in the AIMDS language (McCarty 1980b). The current TAXMAN II system is a frame-based system which uses templates to represent various objects and the relationships which exist between these objects (Cook, et al. 1981).

The TAXMAN system functions by generating instances of the templates in a given context which represent the situation being analyzed. The system also contains semantic descriptions of the possible legal relations among the objects which are represented as variables. The basic process of legal analysis then becomes somewhat of a pattern-matching operation between the various abstractions which are given and the generated expansions. In more simplified terms, the abstract semantic descriptions are expanded until they can be best matched with the given instance of the template. This

description of the analysis process is obviously oversimplified. For a more detailed discussion see McCarty (1977).

The two main problems with the first version of TAXMAN, which the second version attempts to overcome are: 1) the obvious problem that factual descriptions for all conceivable situations which might occur must be represented, and 2) the first system was unable to handle concepts which are open textured or dynamic in interpretation, dependent on the evidence or circumstances present. Despite these problems, when tested on actual corporate tax cases, the results from the TAXMAN project have been encouraging.

The Legal Decision-making System (LDS) is another legal analysis system which has been developed through the implementation of AI techniques. LDS is a rule-based model which uses production rules (antecedent-consequent rules) to represent the process used by an expert to analyze a legal situation. (For a refresher on production rules see Section 2.4.2 in Chapter 2.) LDS is constructed within the ROSIE programming environment which gives the LDS system a readable English-like syntax and the ability to interact with external computer systems (Cook, et al. 1981).

The current version of LDS has been applied to product liability cases. The system is composed of antecedent-consequent rulesets which are used to construct the decision process which a lawyer would use to analyze a case. The rulesets, each representing a legal concept important to the case, act together to produce an effective and comprehensive model of the decision-making process. The two most important advantages of this system are its ability

to explain its own reasoning process and its ability to be modified (Waterman and Peterson 1981). The ROSIE environment, which helps give LDS many of its desirable features, will be discussed in greater detail in Chapter 9.

There are also a few other systems capable of performing rather shallow legal analysis which will be commented on for completeness of this treatise on current legal analysis systems. One such system is JUDITH, a system intended to enhance a lawyer's legal reasoning abilities. Two basic capabilities of the system are important: 1) JUDITH is able to guide a lawyer along various avenues of reasoning through an interactive session during which the lawyer indicates the applicable premises suggested by JUDITH, and 2) JUDITH is also capable of assisting the lawyer in the construction of a hierarchy of premises to support an argument or final premise (Popp and Schlink 1975). Another current project is LEGOL which is intended to build a language suitable for computers which can express legal concepts without losing their meaning (Cook, et al. 1981). For additional information on LEGOL and other current research efforts aimed at applying AI techniques to law see Cook, et al. (1981) and Meldman (1977).

8.4 The Ideal Legal Analysis System

Much speculation has been and is still being made as to what the ideal characteristics of a legal consultation system might be. In this section the characteristics which are most often regarded as being part of the ideal system will be discussed. However, before

discussing them it is necessary to make a few comments regarding the attributes of a law which make it most suitable for computer applications.

Possibly the most desirable attribute of the chosen area of law is a well understood and strict legal interpretation. The area of law selected should also have a depth and complexity which would contribute to its richness of meaning, thereby making it interesting for computerized legal analysis (McCarty 1980a). Other attributes which are desirable are correctness and modifiability (Fiedler 1980). Correctness refers to the exactness of the law and the ability to express this exactness in the context of the system's knowledge base while modifiability refers to the ease with which an area of law can be updated as required.

Once an area of law has been selected for the development of a consultation system the ideal system would most likely exhibit the following characteristics. The knowledge base of the system should contain not only the applicable legal rules, which are contained in the law, but also the text of the law which these rules define. The knowledge base should also contain landmark cases which could be used to back any arguments which might be developed. The system should have the ability to analyze the syntax of the law it contains as well as understand the semantics of this law. From this understanding of the law the system should have the ability to draw analogies from the landmark cases it has access to. In addition to these

cognitive abilities the consultation system must also have a clear understanding of the limits of its area of expertise and be able to handle the concept of time.

Additionally, to facilitate the effective analysis of legal issues in the manner described above, the consultation system must be able to ask pertinent and sophisticated questions to gather the important facts about the case at hand. The system should also be able to explain the reasoning process used to reach a given conclusion. The purpose of this capability is twofold: first to assist in the process of correcting logic and second to educate the user. To help substantiate the reasoning process the system should be able to: 1) list any assumptions which were made at points where incomplete knowledge was encountered, and 2) list crucial decision points accompanied by the decision reached at each of these points and a *corresponding* landmark case which would back this decision.

Finally, the system should be expandable to facilitate changes in the law and to expand the system's area of expertise. The most desirable method of expansion is through experience of the system or through advising sessions which the system would hold with human experts in various areas of law (Niblett 1981, Niblett 1980). Other more obvious characteristics of an ideal consultation system include correctness or "functional equivalence" (Fiedler 1980, p. 144) which refers to the quality of the system's conclusions, efficiency, reliability, maintainability and provability (Fiedler 1980).

8.5 Summary

This chapter primarily addressed previous work which has been done to construct artificially intelligent legal analysis systems. The existing similarities between computer programs and law were discussed. Some of the current legal analysis systems capable of analyzing a case or predicting the outcome of the case were briefly described including SARA, TAXMAN and LDS. Other systems which perform only shallow legal analysis which were mentioned are JUDITH and the LEGOL Project.

This chapter was intended only to give background information pertaining to legal analysis systems without relating any of its content to the system developed by this thesis. Now that sufficient background has been presented in the areas of artificial intelligence, differing site conditions and current computerized legal analysis systems, the work of this thesis can be explained. The following chapters deal specifically with the development, operation and capabilities of the differing site conditions analysis system.

CHAPTER IX

THE PROGRAMMING ENVIRONMENT: ROSIE

9.1 Introduction

As part of this thesis a model has been developed which demonstrates the feasibility of applying AI techniques to contract management. The Differing Site Condition Analysis System (DSCAS) will be discussed specifically in the following chapters. This chapter deals with the ROSIE (Rule Oriented System for Implementing Expertise) programming environment in which the DSCAS model has been developed. The chapter is composed of two main sections. The first presents an overview of the attributes and capabilities of the ROSIE programming environment. The second then examines some of the limitations of ROSIE. Lastly, the summary of this chapter includes a few comments regarding the rationale behind the selection of the ROSIE environment.

9.2 Capabilities of ROSIE

ROSIE is a general purpose programmng environment for the construction of expert systems. The system provides the user with an assortment of functions which can be used in a variety of ways to model intelligent thought processes. Possibly the most significant feature of these functions is their English-like syntax which

produces a comfortable environment for the construction of expert rules. The ROSIE environment also allows the user to build and maintain files, manipulate data bases and interact with remote systems from within the programming environment. (The capability to interact with remote systems, however, is not yet available for versions of ROSIE running on VAX machines.) These and other features will be briefly discussed in this section to give the reader an understanding of the environment in which the DSCAS program was built.

The English-like syntax of the ROSIE environment is an important characteristic of the language giving it a number of desirable features. One of the most important of which is the readability of the code allowing even a novice user to understand many of the rules within a program. This English-like syntax also assists the knowledge engineer in the construction of programs by providing an environment which is adaptable to a wide variety of problem domains and which allows the use of English-like constructs for the expression of an expert's rules (Hayes-Roth, Waterman and Lenat 1983).

The ROSIE environment's English-like syntax is used to store concepts within a data base as well as construct rules for the manipulation of these concepts. The development of concepts within the ROSIE environment is dependent on the establishment of relationships through the use of auxiliary verbs such as; is, was, were, am, will be, does and do. The sentences which express a single relationship through the use of an auxiliary verb are called primitive sentences. Relationships can be either explicitly established by asserting the

relationship or implicitly established by testing for the existence of the relationship. ROSIE is capable of understanding five basic types of relationships (Hayes-Roth, Waterman and Lenat 1983). The following lists each of the allowable types of relationships and an example of each taken from DSCAS.

1.	Class membership	TEST1 is a previous-session.
2.	Prediction	System is restarting.
3.	Intransitive verbs	Notice-requirement was waived.
4.	Transitive verbs	Government did suffer prejudice.
5.	Predicate Complements	Exculpatory-clause is probably valid.

Each of the above relationships can be expanded in a variety of ways (Hayes-Roth, Waterman and Lenat 1983). Each relationship can be negated simply by adding the word "not" at the appropriate place in the sentence, e.g., Government did not suffer prejudice. The relationships can also be placed in the past, present or future tense simply by using the appropriate tense of the auxiliary verb, e.g., Past tense - Difference was material, Present tense - Difference is material, Future tense - Difference will be material. Lastly the relationships can have prepositional phrases added to them to enhance their meaning, e.g., Contractor did suffer prejudice through reliance.

The English-like syntax of the ROSIE environment is so similar to the natural English expression of a concept that one might easily think that ROSIE possesses the ability to understand natural language. This is not true. Although the ROSIE system understands a few words it is NOT a natural-language understanding system.

ROSIE interprets the propositions and rules within a program by concentrating on the grammatical and logical role of words as they relate to one another in the allowable sentence forms, i.e., the ROSIE system pays strict attention to the syntax of a sentence (Hayes-Roth, Waterman and Lenat 1983).

The ROSIE environment also facilitates the representation of a variety of knowledge types through the use of data bases and rules. Data bases can be used to explicitly store relational knowledge in the form of simple sentences. Manipulation of data bases is achieved through a variety of functions such as; "assert" to add a sentence or proposition to the active data base, "deny" to remove a sentence or proposition from the active data base, "go add . . . to unknowns" to add the given proposition to the specified data base (in this example the unknowns data base is specified) and "remove . . . from unknowns" to remove the given proposition from the specified data base. Other functions can be used to remove sentences concerning a specific term, to determine the truth of a proposition within a data base, to display the complete data base or to display just those sentences concerning a specific term. The ROSIE environment allows the use of a global data base as well as any number of specifically designated data bases which can be "private" to a given ruleset if desired (Fain et al. 1981). The DSCAS system uses a number of these functions to maintain the six data bases used to represent the pertinent information.

The second means for knowledge representation is the use of rules. Rules in the ROSIE environment are used to represent three

types of procedural knowledge. The most recognizable type of knowledge is "how-to" knowledge. The "how-to" knowledge is the knowledge of how a lawyer would analyze the differing site condition claim, i.e., the steps in the lawyer's decision process. This knowledge is contained in the rules which designate the order in which the questions are to be asked. Another type of knowledge is classification knowledge. The rules used by the DSCAS program to classify the differing site condition as a Type I or Type II condition exemplify the representation of this type of knowledge. A third type of knowledge is diagnostic knowledge. The DSCAS program contains a simple example of diagnostic knowledge in the rules which are used to decide if the notice requirements were fully complied with. It is important to realize that the procedural knowledge contained by DSCAS's rules is a form of embedded knowledge which is static in nature. Whereas, the knowledge contained in a data base is dynamic in nature experiencing frequent changes caused by actions which have been invoked because of the embedded knowledge.

The rules supported by the ROSIE environment are produced by combining one or more primitive sentences with actions. These actions can be either simple actions such as "send", "assert" and "remove", or compound actions such as "for each," "one of" and "until." Rules constructed within the ROSIE environment can be individual rules or can be organized within a ruleset. Any number of rules and rulesets can be used to construct a program. Since each ruleset is treated as an individual program module by the ROSIE environment

the knowledge engineer is free to construct a system with as much or as little modularity as desired.

A few additional comments concerning the knowledge representation capabilities of the ROSIE environment should be mentioned. A significant advantage of the knowledge representation capabilities of the ROSIE environment is that it allows the knowledge engineer to develop programs which implement pattern directed inference and to a lesser extent productions (See Chapter 3, Section 3.4.1 and 3.4.2, respectively) (Hayes-Roth, Waterman and Lenat 1983). Lastly the ROSIE environment supports the use of eight primitive data types (Kipps 1983). These data types include strings, numbers, names, tuples, class elements, propositions, intensional descriptions and patterns. For additional information on these data types see Fain et al. (1981) and Kipps (1983).

The ROSIE environment also provides the knowledge engineer with a variety of inference mechanisms for the development of sophisticated control strategies. Since ROSIE provides a general-purpose programming environment programs can be constructed which implement state driven, goal driven or change driven control strategies (See Chapter 4, Section 4.1.2) (Hayes-Roth, Waterman and Lenat 1983). A particular control strategy can be implemented by the careful selection and construction of rulesets.

Three specific types of rulesets, procedures, generators and predicates can be constructed in the ROSIE environment. Procedure rulesets are used to represent modular tasks which rely on any number

of parameters (Fain et al. 1981). Procedures can be invoked from within other rulesets using the "go" or "call" action. The invoked procedure will return control to the invoking ruleset when a "return" action or "end" statement is executed. Procedures will in many instances contain the control information for the program. An example of a procedure ruleset taken from DSCAS is shown in Figure 9.1.

Generator rulesets are used to procedurally define a class of elements allowing the system to invoke the ruleset which produces only those elements of a class that meet a specific description. Generators are executed automatically when the description of the element which they produce is encountered in a rule. When the description is executed the appropriate generator is invoked producing the desired elements. An example of a generator ruleset taken from DSCAS is the "To generate a file_to_be_read" ruleset which produces the name of the file which contributed the conclusion pertaining to the no entitlement decision.

Predicate rulesets allow the knowledge engineer to construct rulesets for determining the existence of relationships among elements, thereby eliminating the need to explicitly represent these relationships in the data base. Predicates can also be used to determine the validity of answers to questions posed by the system. A predicate ruleset is invoked whenever the existence of the relationship it defines is in question. Predicate rulesets can only test the truth or falsehood of the relationship and only return the value of "true" or "false." The predicate ruleset shown in Figure 9.2 is used by DSCAS to decide

```
To check-final-payment:

[1] Assert final-payment was checked.

[2] If 'final-payment was made' is true in answers,
      send {3 lines,"The final payment was made.",1 line}
      and activate answers
      and send {1 line,"The date of final payment was ",
                        the date-of-final-payment,".",cr}
      and deactivate
      and deny entitlement is still-probable
      and assert entitlement is not still-probable
      and return,
    otherwise if 'final-payment was not made' is true in answers,
      send {3 line,"The final payment was not made.",1 line},
    otherwise if 'if-final-payment was made' is true in unknowns,
      go unknown-final-payment,
    otherwise,
      go determine-if-final-payment-was-made.

End.
```

Figure 9.1 Procedure Ruleset

To decide answer is valid-answer:

[1] Match the lowercase of the answer:

```
{"act-of-god"}              conclude true;
{"act-of-third-party"}      conclude true;
{"act-of-government"}       conclude true;
{"act-of-contractor"}       conclude true;
{"fault-of-contractor"}     conclude true;
{"fault-of-government"}     conclude true;
{"no-one"}                  conclude true;

default:                    conclude false.
```

End.

Figure 9.2 Predicate Ruleset

if the answer given for the question contained in the "To determine-cause" ruleset is a valid answer. This predicate ruleset is invoked when the statement "if the answer is valid-answer" is executed.

An additional feature which is important to the inference capabilities of the ROSIE environment is the functions used to order the execution of rules within a ruleset. The knowledge engineer can select sequential, cyclical or random execution of the rules within a ruleset. The "execute sequentially" command at the beginning of a ruleset causes the rules to be fired in a top-to-bottom fashion until the last rule is executed at which time a "return" is automatically executed causing the control to return to the invoking ruleset or top level. The "execute cyclically" command causes the same order of execution as sequential execution, however, after the last rule is fired the process begins again with the first rule in the ruleset. This type of control is essential if the rules of a particular ruleset are to be treated as production rules. The "execute randomly" command causes the rules within a ruleset to be fired in a pseudo-random fashion (Fain et al. 1981). A few additional points that should be noted are that if no order of execution is specified the order of execution defaults to sequential execution of the rules. Also, both cyclic and random execution of a ruleset require the execution of a terminate action to return control to the invoking ruleset. For additional information on these and other features of the ROSIE environment see Fain et al. (1981), Hayes-Roth et al. (1981), Fain et al. (1982) and Kipps (1983).

9.3 Limitations of ROSIE

Although the ROSIE environment provides the knowledge engineer with a variety of general-purpose programming capabilities ROSIE does have limitations. Many of these limitations result directly from the desire to keep the ROSIE environment as general as possible. In this section a number of the limitations will be briefly discussed.

A major limitation of the ROSIE environment is that it lacks the ability to directly access its own rules and control mechanisms (Hayes-Roth, Waterman and Lenat 1983). Due to this inability there is no means by which a system built in the ROSIE environment can add to or modify its own structure. Therefore, one cannot construct a system capable of learning new rules or monitoring changes to its own logic. This inability also creates problems when the system is required to explain its line of reasoning.

Although ROSIE provides sophisticated means for accessing information within the data base it lacks sophisticated capabilities for structuring and constructing data bases (Hayes-Roth, Waterman and Lenat 1983). In the current version of ROSIE the sentences within a data base are primitive sentences which are grouped solely by the auxiliary verb within the sentence. Asserted sentences which are more complex are recorded in the data base as a number of primitive sentences. However, related primitive sentences cannot be grouped together and existing relationships scattered throughout a large data base are not readily identifiable. As an example if one were to "assert DSCAS is a legal analysis system that does analyze the

differing-site-condition claim" the primitive sentences added to the data base would look as follows:

DSCAS does analyze CLAIM#1.
DSCAS is a system.
CLAIM#1 is a CLAIM.
DSCAS is legal.
DSCAS is analysis.
CLAIM#1 is differing-site-condition.

If the data base were quite large the only sentences from this assertion which would remain grouped together are those with the same auxiliary verb.

Another limitation of ROSIE is the lack of predefined AI problem-solving strategies. For example ROSIE does not provide any specific means for implementation of operations such as heuristic search or productions. However, since ROSIE is general-purpose some of the AI strategies can be constructed from the capabilities provided by the programming environment. An additional weakness is that the control strategies of systems constructed within the ROSIE environment are relatively inflexible, however, due to the modularity which can be built into a system the effect of this limitation can be significantly reduced (Hayes-Roth, Waterman and Lenat 1983).

Other limitations of the ROSIE environment include limited means to procedurally invoke rulesets and limited memory space (Hayes-Roth, et al. 1981). The current version of ROSIE only supports the computation of the names of procedure rulesets, therefore, generator and predicate rulesets must be invoked by explicitly expressing the invoking command. For example, there is no way to procedurally produce "the file_to_be_read" which then invokes the

appropriate generator. Memory limitations occur because ROSIE is built on top of the INTERLISP language, which requires a large amount of memory (4 megabytes of virtual memory). Memory space only becomes a problem when large systems are constructed within the ROSIE environment. (DSCAS experienced no problem with the amount of available memory.) One last limitation of ROSIE which results in part from limited memory and in part from the INTERLISP environment is that the systems built in the ROSIE environment are quite slow. The Rand Corporation is currently developing a C version of the ROSIE environment which could possibly remedy these speed and memory limitations.

9.4 Summary

This chapter has dealt only briefly with the ROSIE programming environment touching on some of the capabilities and some of the limitations which exist in the current version. The important attributes and capabilities which were discussed include: ROSIE's English-like syntax, the ability to represent a variety of knowledge types within the ROSIE environment and the ability to implement a variety of inference mechanisms. Some of the significant limitations which were mentioned include: the inability of a system constructed in the ROSIE environment to directly access its own rules and control structure, the lack of sophisticated means for constructing and structuring data bases and the lack of predefined AI strategies. Although the current version of DSCAS has not significantly tested

all aspects of the capabilities provided by the ROSIE environment, ROSIE has proved adequate in most aspects required by the current version of DSCAS.

In light of the capabilities and limitations which have been discussed in this chapter a few comments pertaining to the rationale behind the selection of the ROSIE environment for this thesis are in order before concluding this chapter. Due to the time frame placed on this thesis it was realized that a programming environment was necessary to speed the initial development of the analysis system. ROSIE was selected primarily because of the English-like syntax and readability provided by the programming environment. This aspect of ROSIE proved to be quite beneficial by making the learning process more comfortable and somewhat faster than initially anticipated. An additional consideration was that the Legal Decision-making System (LDS), constructed in the ROSIE environment (See Chapter 8, Section 8.3) has previously demonstrated that ROSIE contained the capabilities necessary to perform a form of legal analysis. Lastly, ROSIE was affordable and the computer facilities necessary for its use were readily available. The use of ROSIE was licensed from the Rand Corporation.

CHAPTER X

DEVELOPMENT OF DSCAS

10.1 Introduction

The Differing Site Condition Analysis System (DSCAS) is the first version of an expert system which performs the analysis of the differing site condition claim. The ROSIE programming environment, discussed in the previous chapter, provided the tools for the development of DSCAS. This chapter discusses two important topics: the development of the legal analysis logic and the development of the DSCAS program within the ROSIE environment. The first section discusses how the legal analysis logic on which DSCAS is based was developed. In addition, the rationale behind a few of the important decisions reached during the development process is explained. The second section deals with the development of DSCAS explaining the structure of the system as well as the rationale behind this initial structure. The second section also discusses the function oi the various components which support the core of DSCAS by performing the actual analysis.

10.2 Development of the Legal Analysis Logic

To gain an understanding of how DSCAS performs the analysis of a differing site condition one must first understand the logic

followed during the analysis. Recall, if you will, the discussion of current legal analysis systems contained in Chapter 8. Some of the systems discussed in Chapter 8 implement templates of the law, analyzing the case by matching these templates against the facts of the case. Other systems analyze cases by applying the facts of the case to mathematical formulas which attempt to predict the case based on statistics gathered from previous cases, while others analyze the syntax and semantics of the law applicable to the case at hand then decide the case based on this understanding of the law.

The system developed by this thesis applies a technique which is indirectly based on the combination of the three methods just mentioned. DSCAS employs logic which follows the decision process used by a lawyer in the analysis of the differing site condition claim. This logic is based on the lawyer's understanding and interpretation of the laws which are applicable to the case, as well as on the lawyer's preception of how well certain patterns of fact from the case fit into accepted standards, and lastly, on the lawyer's determination of how decisions from previous cases in a given jurisdiction might support or refute the case at hand.

An important advantage to the use of this analysis technique is that the richness and the meaning of the law understood by the lawyer and incorporated into his reasoning process is retained. This alleviates the problems of giving DSCAS the ability to semantically and syntactically analyze the applicable law. An additional advantage

is that the lawyer's understanding of the court's tendencies is built into his reasoning process, i.e., the lawyer should know which facts a court places special attention on or which facts are treated less formally. An example of this within DSCAS is the treatment of the notice requirement. Although the notice given by the contractor may not meet the notice requirements set forth by the DSC clause, DSCAS does not deny entitlement. Instead, the analysis will proceed to check prejudice to the government. Entitlement is not denied because in the vast majority of cases the courts do not deny entitlement based solely on failure to comply with the notice requirements. This concept could not be deduced from semantic or syntactic analysis of the applicable law.

There are, however, obvious disadvantages to this technique. The greatest disadvantage is that the system has no understanding of the underlying reasons for asking a particular question. This would become especially apparent when a new situation, lying outside the domain of questions asked by DSCAS, is encountered. In such a situation DSCAS would not be able to continue with an accurate analysis. An additional disadvantage is that the method of analysis used by DSCAS is only as good as the analysis procedure used by the experts from which it is derived. In other words, the quality of DSCAS's analysis is highly dependent on the expert's understanding of the claim.

The process used by DSCAS to analyze the differing site condition claim is based on the work of Cobb.[1] Cobb developed a

---

[1] James Cobb, Graduate Student, University of Colorado.

decision tree modelling the DSC analysis process from an intense study of the Board of Contract Appeals (BCA) decisions on DSC claims. During the development of the decision tree the expertise of Manzi[1] was continuously sought to review and refine Cobb's interpretation of the BCA analysis process. The decision tree developed by Cobb implements questions set in the past tense which require only Yes/No answers.

Before the logic developed by Cobb could be implemented in DSCAS a few refinements were necessary. The first of these refinements was to make some of the related Yes/No questions more succinct. This was accomplished in most instances by creating multiple choice questions or by creating questions which could be answered by a few words. In addition, questions concerning the date of an event were added.

The second refinement to Cobb's logic ultimately affected the overall structure of DSCAS. The initial structure of the DSC logic was subdivided into four parts corresponding to Notice, Scope of Clause, Type I and Type II. A quick glance was all that was necessary to realize that too much knowledge was contained in any one of these sub-structures to give DSCAS sufficient modularity. Further breakdown of the logic was necessary to achieve a more favorable level of modularity.

---

[1] Joseph Manzi, Vice President, Kellogg Corporation, Littleton, Colorado. Informal conversations held during March through September, 1983.

Two guidelines were followed to accomplish the breakdown of the logic. First, each module resulting from the breakdown should be concerned with only a small number of related concepts and second each module should have a minimal number of paths for ingress and egress. The first guideline was implemented to simplify the logic within each module and to create modules which could be incrementally developed and tested, thus simplifying the development process.

The second guideline was implemented to help create modules which had a minimal number of interrelationships. The effect of this guideline was to reduce the complexity of the overall control structure and to reduce dependencies between modules. By reducing dependencies between modules separate groups of concepts could be tested independently, i.e., individual modules which compose the analysis can be run independent of the complete analysis.

The end result of breaking down the logic was to produce twenty-two separate modules which represent the underlying logic of the DSC claim. The individual modules are as follows:

DSC Assertion	Latent Deficiencies
Final Payment	Standard Conditions
Notice Form	Site Inspection
Notice Promptness	Superior Knowledge-I
Responsible Receiver	Superior Knowledge-II
Government Prejudiced	Reliance I
Contract Obligation	Reliance II
Excluded Conditions	Material Differene-I
After-Bid Conditions	Material Difference-II
Express-Implied Conditions	Exculpatory Language-I
Contract Indications	Exculpatory Language-II

Diagrams depicting the analysis logic contained by each module as well as the overall flow diagram which depicts how the individual

logic modules are linked to peform the complete analysis of a DSC claim can be found in Appendix A.

One additional point concerning the decision process should be mentioned. In the process of selecting and phrasing questions used to gather information for the decision process emphasis has been placed on the development of questions or sequences of questions which rely more on technical knowledge than on legal knowledge. The reason behind this is that in the vast majority of cases field personnel have a great deal more technical knowledge than legal knowledge. The success of this attempt, however, has been somewhat limited as can be seen in the number of questions which still rely on legal judgment.

10.3 Structure of the DSCAS Program

Following the development of the logic and an initial refinement period for that logic, work was begun on construction of the DSCAS program. DSCAS is dependent on six major components, each of which is composed of one or more files. These six components include: the driver rulesets, the question rulesets, the entitlement rulesets and the top-level control and other peripherals. The purpose and development of each of these components will be discussed in the following sub-sections.

Before discussing the various components in detail it is necessary to present an overall view of the control process in DSCAS. Figure 10.1 depicts the flow of control and the relationships between the major components of the DSCAS program. The session begins

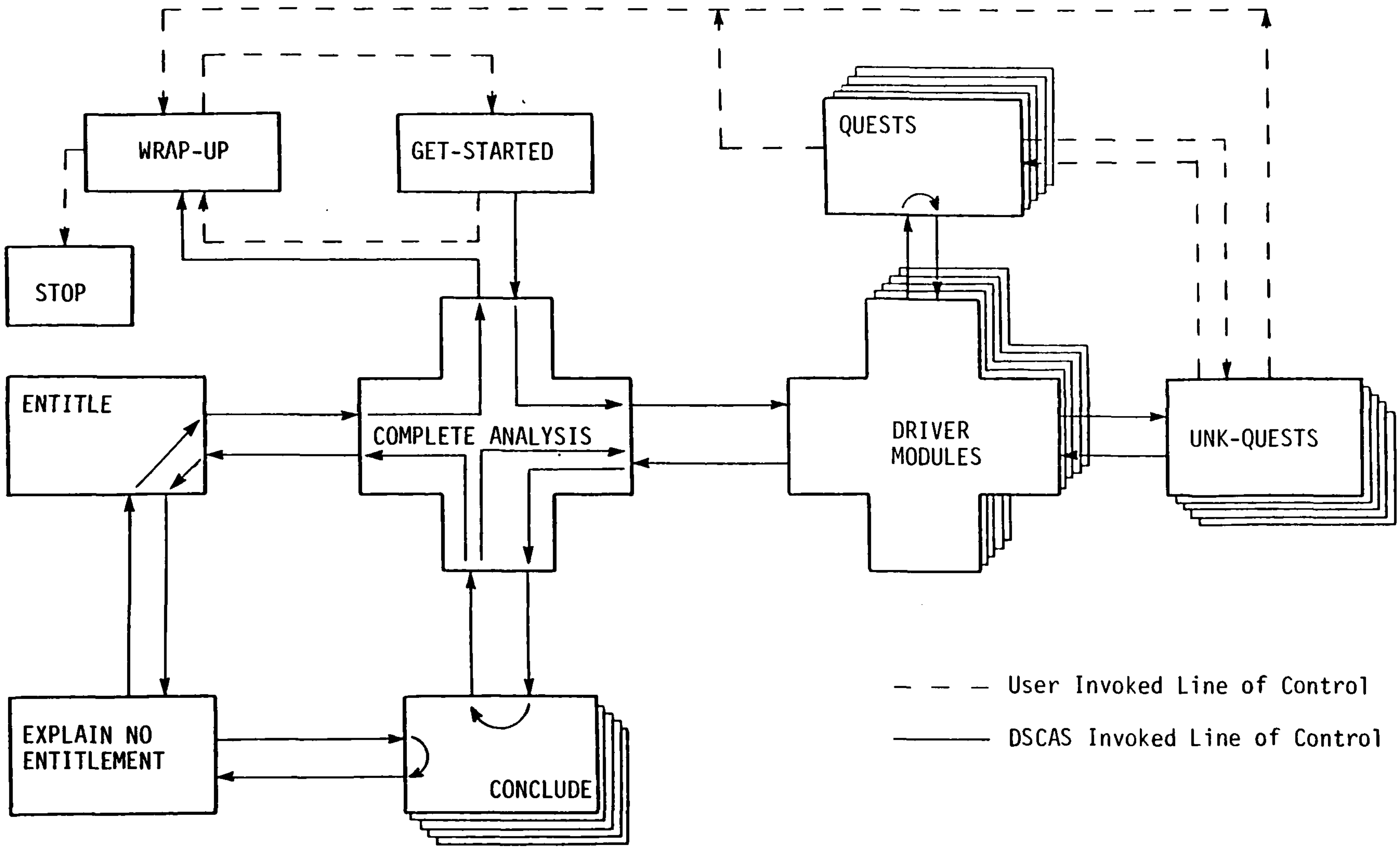


Figure 10.1 DSCAS Control Diagram

at the GET-STARTED module which sets up the initial environment essential for the analysis. Control then automatically is passed to the selected DRIVER MODULE. In this case, the COMPLETE ANALYSIS module has been selected. At this point the analysis portion of the session begins.

The COMPLETE ANALYSIS module begins by invoking the DSC Assertion DRIVER module. The DSC Assertion module is then executed, testing for known information. If nothing pertinent to DSC assertion is known, the appropriate QUESTS ruleset is invoked which requests information about the DSC assertion from the user. If it is discovered that the DSC assertion information was unknown during a previous session, the appropriate UNK-QUESTS ruleset is executed which checks to see if any new information has been gained since the last session. The control follows a cyclic pattern between the invoked DRIVER module and the QUESTS and UNK-QUESTS files until either a "return" or end statement returns control to the COMPLETE ANALYSIS module which invokes the appropriate CONCLUDE ruleset. The invoked CONCLUDE ruleset then draws any pertinent conclusions based on the answers given to questions asked by the previously invoked DRIVER module. After all conclusions have been drawn, control again returns to the COMPLETE ANALYSIS module. At this point if it has been concluded that a "reason for no entitlement does exist," control is passed to the ENTITLE module. Otherwise, the next appropriate DRIVER module is invoked and the cycle begins again.

When the ENTITLE module is invoked, the ruleset first determines which conclusion has been drawn indicating no entitlement. Once this conclusion has been located, the EXPLAIN NO ENTITLE module constructs the explanation for the conclusion. The EXPLAIN NO ENTITLE module constructs the explanation by returning to the appropriate CONCLUDE file and testing the various rules to determine which rule drew the conclusion being explained. Once the rule is located, it is read and a copy of it is carried back to the EXPLAIN NO ENTITLE module where it is dissected to find exactly which statements within the rule are true. After displaying an explanation for the conclusion, control is returned to the ENTITLE module which cleans up the databases, and closes any files still open and then returns control to the COMPLETE ANALYSIS module. Since it has been concluded that no entitlement is possible, the COMPLETE ANALYSIS module passes control on to the WRAP-UP module via the GET-STARTED module.

The WRAP-UP module then wraps up the session by allowing the user to view the results of the session and store the session if desired. Once the session has been wrapped up, the user is asked whether or not he wishes to perform another analysis. If the response is "yes," control is passed back to the GET-STARTED module which begins the complete process over again. If the response is "No," control is passed to the top-level, i.e., control is passed from DSCAS back to the ROSIE environment.

A few additional comments pertaining the Figure 10.1 are necessary to fully understand it. As one can see the lines of control are both solid and dashed. The solid lines represent lines of control automatically followed by DSCAS while the dashed lines represent user invoked lines of control. The means for user invocation of these lines will be explained in detail in later sub-sections. In addition the lines of control drawn inside the modules attempt to clarify the viable control routes within a module, i.e., the CONCLUDE module can only return control to the invoking module whereas the COMPLETE ANALYSIS module can pass control to a number of different modules dependent on the current state of the system.

10.3.1 Control of the Analysis Process

Control of the analysis process utlimately lies in the rulesets of the DRIVERS files. These rulesets are procedures which define the steps during the analysis process. There are twenty-two control rulesets, one corresponding to each of the logic modules listed in the previous section. In addition there is a ruleset for the control of the complete analysis which calls up each of the other twenty-two rulesets individually at appropriate times during the anlysis. The rulesets are divided between the three DRIVERS files in the following manner:

DRIVERS	- DSC Assertion Final Payment Notice Form Notice Promptness	Responsible Receiver Government Prejudiced Complete Analysis
DRIVERS2	- Contract Obligation Excluded Conditions	After-Bid Conditions
DRIVERS3	- Express-Implied Conditions Contract Indications Latent Deficiencies Standard Conditions Site Inspection Superior Knowledge-I Superior Knowledge-II	Reliance I Reliance II Material Difference-I Material Difference-II Exculpatory Language-I Exculpatory Language-II

The division of rulesets between DRIVERS files is based on Cobb's initial setup of the DSC logic, i.e., DRIVERS contains the form of notice logic, DRIVERS2 contains the scope of clause logic and DRIVERS3 contains the Type I and Type II logic.

The rulesets which control the analysis process for each logic module are all constructed in a similar fashion. An example of a control ruleset, taken from the DRIVERS file, is shown in Figure 10.2. The "To check-final-payment" ruleset is used to determine what information is known concerning the final payment.

The reason for constructing all rulesets controlling the analysis process in a similar fashion is two-fold. First, using a similar format for all control rulesets produces an overall control structure which is relatively simple in nature and is therefore easier to debug. The second reason for the use of a similar format is the resulting uniformity. Uniformity is necessary if the system is to be able to work with its own rules, e.g., if rules are similar in nature, a generic ruleset can be developed to read all rules of a similar format.

```
To check-final-payment:

[1] Assert final-payment was checked.

[2] If 'final-payment was made' is true in answers,
      send {3 lines,"The final payment was made.",1 line}
      and activate answers
      and send {1 line,"The date of final payment was ",
                        the date-of-final-payment,".",cr}
      and deactivate
      and deny entitlement is still-probable
      and assert entitlement is not still-probable
      and return,
    otherwise if 'final-payment was not made' is true in answers,
      send {3 line,"The final payment was not made.",1 line},
    otherwise if 'if-final-payment was made' is true in unknowns,
      go unknown-final-payment,
    otherwise,
      go determine-if-final-payment-was-made.

End.
```

Figure 10.2 Control Ruleset for Final Payment Logic Module

There are basically two types of rules in every control ruleset except the ruleset controlling the complete analysis (which will be discussed separately later in this section). The first rule executed in each control ruleset simply makes an assertion to the global data base. This assertion is a form of meta-knowledge which allows DSCAS to keep track of the modules which have been executed and to relocate the control ruleset being executed when the anlaysis is interrupted by the user. An example of this type of rule is rule one in Figure 10.2.

The second type of rule found in the control rulesets are procedures used to determine what is or is not known about a particular concept. These rules are the most common rules used in the control rulesets. The first part of these rules is a set of constraints which determine if the remaining portion of the rule should be executed by checking to see if the given propositions are true in the ANSWERS data base. If the sentences do exist, the remaining portion of the rule will be executed. If not, the next rule is examined in a similar fashion. The method of controlling the execution of rules based on previous answers gives DSCAS the ability to simulate the logic of a particular module. If no propositions are to be tested, as in the example shown in Figure 10.2, the rule is executed automatically.

The last part of the procedural rule is used to determine what specific information, if any, is known about a certain concept. This is accomplished by testing to see if any valid answer has been given previously and currently exists in the ANSWERS data base. If

a valid answer does exist in the ANSWERS data base, a short statement is sent to the user to inform him of the known information. This information serves two purposes: first, it provides valuable information for debugging the logic modules and, second, it gives the user an awareness of the state of the analysis. If no valid answer exists, DSCAS then checks to see if the answer to the question was unknown during a previous session and, if so, asks the user if he now knows the answer to the question. If none of the tests prove true, DSCAS asks the user a question pertaining to the concept.

The complete analysis control ruleset serves a different purpose than the other control rulesets and is therefore structured differently. The most important difference is that it executes cyclically until the analysis is complete, thereby continuously checking the state of the system by monitoring which logic modules have been checked and whether or not a reason for no entitlement has been found. The complete analysis module also orders the invocation of other control rulesets which are part of the overall control of the analysis process based on the current state of the system and the conclusions which have been reached during the analysis.

The control process used by the complete analysis module is set up as a series of rules, each monitoring the eligibility of one or more logic modules. When a rule is executed, it first determines whether or not the logic module which it controls has been executed. If the logic module has not been checked, the rule then determines whether or not entitlement is still probable. If entitlement is not

still probable, control continues on to the next rule. If entitlement is still probable, additional constraints are checked. These additional constraints are based on conclusions which should have been drawn during the execution of previous logic modules or are based on what other logic modules have been previously executed. The additional constraints are responsible for the precise order of execution of the logic modules.

If all the constraints pertaining to a particular logic module are met, the appropriate control ruleset is invoked. This in turn invokes the necessary question modules. Once control has been returned to the complete analysis module, it invokes the appropriate ruleset for drawing the conclusions based on the answers given to questions asked by the logic module. The conclusions which are drawn are then used to determine which logic module is to be invoked next, beginning the cycle over again.

The complete analysis module will continue execution until one of two states is reached. Either DSCAS will conclude that a reason for no entitlement does exist, in which case the reasoning will be explained prior to stopping the analysis or the analysis will be completed, i.e., all appropriate logic modules will be executed and no reason for denial of entitlement will be found, in which case the system discontinues the analysis session. When the execution of the complete analysis module ceases, control returns to the invoking ruleset or the GET-STARTED ruleset. One final note is that in the current configuration most analyses will cause the complete analysis

module to execute only once through rather than in the cyclic fashion which it is capable of. A copy of the text of the DRIVERS files is included in Appendix B.

10.3.2 Questions Pertaining to the Analysis

The questions which pertain directly to the analysis process are contained in two sets of files, the QUESTS files and the UNK-QUESTS files. There are three QUESTS files. The questions contained in each file are invoked by the control modules of the corresponding DRIVERS files, i.e., QUESTS2 and UNK-QUESTS2 contain all questions which can be invoked by the control rulesets in the DRIVERS2 file. Each of the QUESTS and UNK-QUESTS files contain a number of rulesets, one ruleset for each question which can be asked by DSCAS. However, there are a few exceptions where a ruleset is set up to ask the same question in two different ways depending on the known information at the time. The QUESTS files contain the questions which concern specific information about the DSC while the UNK-QUESTS files contain questions for determining if an answer which was previously unknown is known yet.

There are three different types of questions which can be asked by DSCAS to gather pertinent information. The most common type are Yes/No questions. These questions are quite simple and are phrased so that the user can answer with either a "yes" or a "no." Other viable answers to this type of question are: "-," if the answer to the question is unknown, "??," to get an explanation of the

logic behind asking the question (Note: This option is not yet functional.) or "q," if the user desires to quit the analysis session. The "To determine-government-control" ruleset shown in Figure 10.3 is an example of a Yes/No question ruleset implemented by DSCAS.

All of the question rulesets are set up in a similar fashion. Rule one of the ruleset displays the question and sends control to the appropriate ruleset which is used to read the answer entered by the user. The "To input-yes-no" ruleset called by the question ruleset shown in Figure 10.3 is called by all Yes/No questions in the QUESTS files. This ruleset is used to determine valid answers and to direct the control if responses, such as: "q," "?" or "??" are given. The qnum term used in the first rule is the question number which is displayed with each question. This number is incremented after each question is answered; therefore, it is not fixed for any one question.

The second rule in most question rulesets is used to execute the appropriate actions based on the response given by the user. These rules only need to be set up to handle valid answers since all responses are screened by the ruleset used to read them. In most cases the only action taken in the second rule is the addition of new information to one or more of the databases. Also, in some instances these rules will send information to the user which often pertains to a conclusion which has been drawn from the answers. Lastly, the return command has been added to the question ruleset to increase execution speed, i.e., the control returns to the invoking ruleset at

```
To determine-government-control:

[1] Send {2 lines,"[",the qnum,"]",2 blanks,
              "Did the government have control over the actions",cr,
              "of the third party which is at fault? (Yes/No)",cr,": "}
     and assert dgc is a question
     and go input-yes-no
     and let the qnum be (the qnum + 1).

[2] If the lowercase of the string = "yes",
     go add 'government did have control over third party' to answers
     and return,
   otherwise if the lowercase of the string = "no",
     go add 'government did not have control over third party'
                                                   to answers
     and return,
   otherwise if the string = "-",
     go add ' if-government did have control over third party'
                                                   to unknowns
     and go add 'government did not have control over third party'
                                                to assumptions.

End.
```

Figure 10.3 Yes/No Question Ruleset

the execution of the "return" rather than waiting until the end statement is executed.

The two other types of questions which DSCAS is capable of asking are multiple choice questions and questions which can be answered with a word or phrase. The rulesets for these types of questions are similar in structure to the Yes/No questions; therefore, examples will not be shown. However, a few comments are in order. The answers to the multiple choice questions can be any of the variables listed as well as "-," "?" or "??." The "-" and the "??" have the same meaning here as in the Yes/No questions; however, the "?" can now be used if the user desires an explanation of how to answer the question. The questions which can be answered with a word or phrase can also be answered with "-," "?" or "??" as well as the valid responses pertaining to the question.

All of the questions contained in the UNK-QUESTS files are Yes/No questions that are used only to check if an answer to a question previously unknown is know yet. The rulesets contained by the UNK-QUESTS files are invoked only when a previous session is being rerun and the response to a question during that session was a "-." Also, in rare cases, questions from the UNK-QUESTS files will be asked when a session is restarted after a user invoked interrupt. This will occur only when a question in the module which is being restarted was answered with a "-" prior to quitting the analysis. The questions in the UNK-QUESTS files are NOT used to answer the specific question which pertains directly to the analysis. The

question pertaining to the analysis will be asked if the response to the questions asked about the unknown information is "Yes." The "To unknown-government-control" ruleset shown in Figure 10.4 is an example of a question about an unknown answer.

The ruleset shown in Figure 10.4 is the UNK-QUESTS ruleset which corresponds to the QUEST ruleset shown in Figure 10.3. As can be seen the structure of the UNK-QUESTS rulesets are quite similar to the question rulesets in the QUESTS files. The major difference in the first rule of the ruleset is the ruleset which is called to read the response to the question. The UNK-QUESTS rulesets all invoke the "To read-yes-no" ruleset which reads the response and allows only "Yes," "No" or "?" responses, i.e., these questions cannot be answered with a "-." The actions taken in the second rule of the ruleset are self-explanatory. The knowledge of an unknown response is removed if the answer is "Yes" and, if the answer is "No," an assumption pertaining to the analysis question is made. This process rebuilds a copy of the assumptions data base which would have been created when the unknown answer was initially given.

A few additional comments are necessary before leaving this discussion of the questions pertaining to the analysis. It is important to remember that all responses to questions are screened to determine if the response is a valid response for the question. This feature protects DSCAS from being fooled by an invalid response. Also, each question is currently assigned an identity code, usually the first letter of each word in the ruleset header. This identity

```
To unknown-government-control:

[1] Send {2 lines,"Do you know yet whether or not the government did ",cr,
              "exercise control over the actions of the ",cr,
              "third party? (Yes/No)",cr,": "}
     and assert dgc is a question
     and go read-yes-no.

[2] If the lowercase of the string = "yes",
     go remove 'if-government did have control over third party'
                                          from unknowns
     and go determine-government-control
     and return,
   otherwise if the lowercase of the string = "no",
     go add 'government did not have control over third party'
                                          to assumptions.

End.
```

Figure 10.4 Question Ruleset for a Previously Unknown Answer

code is not currently used; however, it is intended for use when the module for the explanation of the logic behind asking each question is constructed. The text for each of the QUESTS and UNK-QUESTS files can be found in Appendix B.

10.3.3 Data Bases Within DSCAS

DSCAS implements six data bases to store knowledge gathered during the analysis. The knowledge stored in the data bases ranges from user responses to conclusions which have been drawn during the analysis to knowledge concerning the state of the analysis. The data bases are used to represent dynamic knowledge pertinent to the analysis. The six data bases used by DSCAS are: the ANSWERS data base, the UNKNOWNS data base, the ASSUMPTIONS data base, the REPORTS data base, the GLOBAL data base and the STATUS data base. Each of these data bases is used to represent a specific type of knowledge.

The ANSWERS data base is used to represent the information obtained from the user's response to each question. Since a user gives answers consisting of only a Yes/No, a variable or a single word or phrase to each question, DSCAS must record this response in a meaningful way. To do this the response is transformed into a statement which is specific to the answer given and the question asked by the question ruleset, e.g., the response "Yes" might be recorded by asserting "government did have control over third party" to the ANSWERS data base (see Figure 10.3). The purpose of the ANSWERS data base is to represent this transformed answer as a statement which can be used for future reference.

The ANSWERS data base is built incrementally as the analysis progresses when a session is initially run. If the user stores the results of the session, a copy of the ANSWERS data base is recorded using the "dump as (session name)" command which creates a file named (session name). DATABASE, e.g., if the session name is TEST1, the file containing a copy of the ANSWERS data base is called TEST1. DATABASE. Then, when the user desires to rerun the analysis of a previous session, the ANSWERS data base built during the previous session is restored using the "restore (session name)" command. Once the ANSWERS data base has been restored, the analysis progresses by testing for answers which are true in the ANSWERS data base from the previous session. During the rerunning of a session the ANSWERS data base can be added to if questions are encountered which were previously unanswered or answered with unknown.

The UNKNOWNS data base is used to represent the information which is unknown during the analysis session. To create an assertion that is added to the UNKNOWNS data base a question is answered with a "-." The structure of sentences added to the UNKNOWNS data base is unique. The sentences are intended to mean more to the user than to DSCAS. An assertion to the UNKNOWNS data base might look as follows: "if-government did have control over third party" (see Figure 10.3). Obviously DSCAS has no idea what an "if-government" element is; however, the user can read this sentence and know exactly what information is unknown. DSCAS only uses the sentences in the UNKNOWNS data base to determine if the answer

to a question was previously unknown. This test is performed in the control rulesets.

DSCAS uses the UNKNOWNS data base in the same fashion as the ANSWERS data base is used, i.e., to store the results of a session and to reconstruct a data base from a previous session. When a session is stored, the UNKNOWNS data base is stored in a file named U-(session name).DATABASE. Also, when a previous session is rerun, the information contained in the UNKNOWNS data base is used only to invoke the questions contained in the UNK-QUESTS files. The ANSWERS data base and the UNKNOWNS data base are the only data bases stored if a session is stored. DSCAS is constructed so that the information in the remaining data bases will be reasserted as the session is rerun.

The ASSUMPTIONS data base is used in conjunction with the UNKNOWNS data base to represent any assumptions which are made by DSCAS whenever a question is answered with unknown. Assumptions are used by DSCAS to allow the analysis to continue even though an answer might currently be unknown. The assumptions are added to the ASSUMPTIONS data base either when unknown is the response to a question or when the user responds with a "No" answer to a question from the UNK-QUESTS files. The ASSUMPTIONS data base is built incrementally during the analysis and is not stored at the end of the analysis session. In addition, it should be noted that the assumptions which are currently made are the assumptions which will allow the analysis to continue and are not made on any other basis.

Also, it is important to remember that information contained in the ANSWERS, UNKNOWNS and ASSUMPTIONS data bases is used by the control rulesets to determine the order in which questions within a logic module should be asked.

The REPORTS data base is used to represent all conclusions which DSCAS draws based on the answers given to the questions. Conclusions are currently added to the REPORTS data base by rulesets in the CONCLUDE files, by rulesets in the QUESTS files and by rulesets in the DRIVERS files. In the near future DSCAS should be restructured to allow all conclusions to be drawn in the CONCLUDE files. The REPORTS data base is built incrementally as the analysis progresses and is not stored at the end of the analysis. The conclusions in the REPORTS data base are used primarily by the COMPLETE ANALYSIS module to determine the order in which the control rulesets for the various logic modules should be invoked and to draw associations between a number of related answers.

The GLOBAL data base serves as the means for representing all information necessary for keeping track of the current state of DSCAS, i.e., the GLOBAL data base is used as a blackboard by the DSCAS program to monitor itself. The GLOBAL data base is used to represent meta-knowledge, such as which logic modules have been executed and whether or not the analysis has been stopped by the user. This type of knowledge can then be used to return control to a particular logic module if the session is to be restarted after a user invoked interrupt. The GLOBAL data base also temporarily stores the answer to the latest question while its validity is being

checked. In addition, the GLOBAL data base keeps track of the values of the various counters, such as the question number and the number of questions answered with unknown. The GLOBAL data base is built in part by reinitializing it at the start of each session and asserting some initial information and in part by assertions made throughout the analysis session.

The STATUS data base is used by DSCAS to keep track of information concerning any previous sessions. This data base records the names of all sessions which have been stored in (session name). DATABASE files, the author and the date of creation of these files. The STATUS data base is accessed when the user desires to load a previous session. When a previous session is to be loaded, it is accessed for all the names of previous sessions from which the user may choose. The STATUS data base is also accessed when DSCAS requests the author's name and date of creation of the previous session about to be run. Assertions are only added to the STATUS data base when a session is to be stored, at which time the session name, author of the session and date of the session are added to the STATUS data base. The STATUS data base is automatically restored at the beginning of each session and stored at the end of each session.

Before leaving this discussion of DSCAS's use of data bases it is necessary to mention a few additional points. First, DSCAS currently uses six different data bases. Initially this was thought to be best because the various types of knowledge could be recorded separately and used in different applications without problems of

misunderstood pieces of knowledge. At present, however, it appears that the use of six data bases has proved too cumbersome and inefficient in the retrieval and manipulation of the knowledge contained in the data bases. This problem warrants further study to determine the optimum number of data bases.

An additional factor contributing to the problems of retrieval and manipulation of data base knowledge is the structure of the propositions. At present all propositions asserted to all the data bases are simple sentences with some having propositional phrases attached to them. Currently the propositions are quite readable and easily understood; however, they do not allow DSCAS to fully implement the power of the ROSIE environment. The initial attempt when constructing these propositions was to assert sentences into the data base which were *self-contained* and easily understood. This conciseness kept the data bases from becoming cluttered with large numbers of related primitive sentences which in turn provided very readable data bases when displayed. This problem also warrants further study to increase the implementation of the capabilities of the ROSIE environment.

10.3.4 Explanation capabilities of DSCAS

The current version of DSCAS possesses the ability to explain the basis for concluding that a "reason for no entitlement does exist." The explanation capabilities of DSCAS are contained in the UTILS file while the explanation itself is created from information contained in the ENTITLE file and the CONCLUDE files.

DSCAS's explanation module is automatically invoked by the COMPLETE ANALYSIS module when it has been concluded that "entitlement is not still probable." The explanation module will also be invoked by the GET-STARTED ruleset if a single logic module has been run which concluded that "entitlement is not still probable."

The process of explaining the reason for concluding that no entitlement will be allowed begins when an answer is given which causes DSCAS to suspect a reason for no entitlement might exist. This occurs when "entitlement is not still probable" is asserted into the GLOBAL data base and control is returned to the invoking ruleset. At this point DSCAS is not completely certain that a reason for no entitlement exists; therefore, all appropriate conclusions are drawn and control is passed on to the "To check-entitlement" ruleset. This ruleset then determines if a reason for no entitlement does indeed exist. If no reason exists, control is returned to the next designated logic module. On the other hand, if a reason for no entitlement does exist, this is asserted into the REPORTS data base and the actual explanation process begins.

To explain the reason for concluding that no entitlement is probable, DSCAS opens the ENTITLE.TEXT file and begins extracting conclusions which could cause entitlement to be denied. The conclusions are read one at a time and tested until one is found to be true. This conclusion is then matched against the list of possible conclusions in a generator ruleset to generate the name of the file which drew the conclusion. Once a file has been located and opened

to read, DSCAS begins extracting the rules within that file, testing each to determine if it fired. When the correct rule is located, a copy of the rule is made and the file from which the rule was read is closed. The rule is then broken down by pulling off the propositions it contains one at a time. These propositions are then tested in the appropriate data base. Statements which test true are displayed to the user so he can determine precisely why the rule fired. Figure 10.5 is an example of what an explanation of no entitlement might look like.

The current explanation capabilities produce the correct explanation; however, the explanation is extremely slow and needs to be expanded to other areas of the analysis. The explanation is slow because of the read, bind and test cycle used to locate the appropriate conclusions, rule and statements. A single explanation requires between five and ten minutes depending on the system load. The explanation capabilities should also be expanded to other areas of the claim, such as explanation of the logic behind asking a particular question. This can be accomplished quite easily because the current explanation apparatus was set up with this expansion in mind. The expansion of the explanation capabilities will be discussed in greater detail in Chapter 11. The text of the rulesets for the explanation process is contained in the UTILS file which is included in Appendix B.

```
Thinking.......

**** Entitlement will probably not be allowed under DSC claim. ****

I believe that entitlement will probably not be allowed because I
concluded that the exculpatory-clause is probably valid.

The following rule proved to be true causing me
to draw the above conclusion:

If
('exculpatory-clause is specific-to DSC clause'
              is true in answers
     or 'exculpatory-clause is specific-to DSC clause'
              is true in assumptions)
     and ('exculpatory-clause is clear-and-unambiguous'
              is true in answers
      or 'exculpatory-clause is clear-and-unambiguous'
              is true in assumptions)
   then the exculpatory-clause is probably valid.

The following statements contained by the rule
are true:

[1] exculpatory-clause is specific-to DSC clause   (in answers)

[2] exculpatory-clause is clear-and-unambiguous   (in assumptions)
```

Figure 10.5 Explanation of No Entitlement

10.3.5 Peripherals

Before ending this discussion of the DSCAS program, it is essential to highlight some of the peripheral rulesets important to the overall operation of the system. The peripheral rulesets are used indirectly during each analysis to perform the background operations which the analysis relies on. These operations include: reading input, testing the validity of input, loading previous sessions, storing current sessions and starting and wrapping up the analysis, to name a few. All of the peripheral rulesets are contained in either the STARTUP file or the UTILS file.

The "To get-started" ruleset is ultimately the top-level of the DSCAS program and is invoked when the "Go get-started" command is given in the ROSIE environment (after the STARTUP file has been loaded). It is this ruleset to which the control returns when the analysis is complete. The "To get-started" ruleset takes care of loading the files necessary to the analysis, if they are not yet loaded, and creation of the data bases essential to the analysis, if they are not yet created. This ruleset also allows the user to choose between running a previous session or beginning a new session. If the analysis was stopped by the user, this ruleset also restarts the analysis process at the appropriate logic module. Two additional functions of this ruleset allow the user to set up a file which records a transcript of the session and to select the analysis which is desired, i.e., which logic module is to be invoked for this session.

The "To wrap-up" ruleset is invoked by the "To get-started" ruleset once the analysis is complete. The "To wrap-up" ruleset takes care of wrapping up the session by allowing the user to examine the results of the session and storing the session if desired. This ruleset also closes the transcript file if one exists and, if the session was an original, i.e., not a rerun of a previous session, the transcript file is automatically copied to a (session name).ORIG file. This added feature allows the user to keep copies of reruns of the session without deleting the transcript of the original session. The last function of the "To wrap-up" ruleset allows the user to start a new session, if desired, without exiting from the DSCAS environment. This feature is accomplished simply by returning control to the "To get-started" ruleset.

The "To get-started" ruleset and the "To wrap-up" ruleset are contained in the STARTUP file while the remaining peripheral rulesets are contained in the UTILS files. The first set of rulesets in the UTILS files is used to read input for the questions asked by DSCAS. These rulesets execute cyclically until a valid answer is given which causes explanation rulesets to be invoked or control to be returned to the ruleset which asked the question. Explanation rulesets will be invoked if either a "?" or a "??" are given as answers to questions. Once the explanation has been given, the user is prompted for an answer. If a "q" is given, the analysis is interrupted and control is passed to the "To wrap-up" ruleset. There is a ruleset for each of the following types of input: dates, Yes/No with

unknown option, Yes/No without unknown option, multiple choice answers and Before/After answers.

A few comments should be made concerning the specific characteristics of these rulesets. The current ruleset for reading dates checks only the validity of the month entered; therefore, any day and year will be accepted. This should be fixed to be certain that the month contains the day specified. There are two rulesets for reading input to Yes/No questions to facilitate the use of the unknown response. It was determined that the use of two separate input rulesets would be more efficient than keeping track of which questions could not be answered with a "-." The ruleset which reads answers for multiple choice questions is capable of discerning whether or not the variable input is a valid choice for the questions, e.g., a question with possible answers of "a" through "d" cannot be answered with an "f." This feat is accomplished by having each question assert the number of possible variable responses into the data base. When the ruleset to read the response is invoked, it tests the response as well as the range of viable responses to be certain that the response is within the allowable range of choices.

The next group of rulesets in the UTILS file is used to test the validity of answers to questions which receive one or more words in their response. These rulesets are predicate rulesets which conclude whether or not the response given matches a viable choice. There are three of these rulesets: the first is for determining the validity of the month when a date is entered, the second determines

the validity of the response to the question used to determine the cause of the DSC and the third is used to determine the validity of the logic module selected for the analysis.

The remaining rulesets in the UTILS file are used to perform a variety of functions ranging from loading and storing the results of a session to displaying the results of a session. These rulesets are invoked only by the "To get-started" and "To wrap-up" rulesets when a session is beginning or concluding. A few other miscellaneous rulesets contained in the UTILS file perform operations, such as creating and clearing the data bases and counting the number of questions answered with unknown. The UTILS file also contains the rulesets which compose the module for explaining no entitlement. The text of the STARTUP and UTILS files can be found in Appendix B.

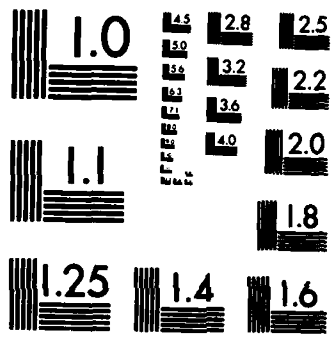
10.4 Summary

This chapter has dealt with the development of the DSCAS program and the underlying reasons for the various features and capabilities of the current version. Throughout the discussion comments have been added concerning particular features which could or should be improved on in future work. The chapter was divided into two main sections: the first discussing the development and refinement of the logic on which DSCAS is based and the second discussing the structure of the DSCAS program.

The discussion of the development and refinement of the logic on which DSCAS is based covered such pertinent topics as how

AD-A146 681 THE APPLICATION OF ARTIFICIAL INTELLIGENCE TO CONTRACT MANAGEMENT(U) CONSTRUCTION ENGINEERING RESEARCH LAB (ARMY) CHAMPAIGN IL T A KRUPPENBACHER AUG 84 3/6

UNCLASSIFIED CERL-TM-P-166 F/G 9/1 NL



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

the logic was structured and the advantages and disadvantages of using a lawyer's decision process as the model for the logic structure. Also, this section explained the changes which were made to the initial logic structure and the reasons for these changes.

The second section explained the structure of the DSCAS program by first presenting the overall structure of the DSCAS program and then explaining the major components of the program essential to the analysis of the claim. The various components which were discussed include the control structure, the question modules used by DSCAS to gather essential information, the use of data bases within DSCAS to represent the information currently known, the explanation capabilities possessed by DSCAS for the explanation of the reason for no entitlement and the peripherals used to support the overall operations of the DSCAS program.

Although this chapter presented a thorough explanation of the structure and internal operation of the DSCAS program, as further clarification, a user's manual containing a sample session has been included in Appendix E.

CHAPTER XI

DSCAS IN PERSPECTIVE

11.1 Introduction

The overall performance of the DSCAS program is examined in this chapter by analyzing the test results. As part of the appraisal of DSCAS's performance the accomplishments as well as the limitations encountered during each test case will be discussed. Also, the method of selecting test cases and the actual testing process will be briefly explained. Based on the test results some conclusions are drawn regarding the overall work of this thesis and the performance of DSCAS. Then finally, a number of recommendations for further study and development of the DSCAS program are presented.

11.2 DSCAS Test Results

The results of testing to determine DSCAS's abilities were quite encouraging. However, a number of limitations also became evident. This section discusses the testing of DSCAS and the implications of the test results. Before delving into a discussion of DSCAS's performance on actual test cases, it is necessary to explain briefly how the test cases were selected and how they were run.

To begin the selection process thirty-seven BCA cases were selected from the list shown in Appendix D which has been compiled

from cases referenced in Cibinic and Nash (1981) and Currie et al. (1971). These thirty-seven cases were selected on the basis of the major concepts pertaining to the DSC claim with which they were concerned, i.e., some dealt with notice and prejudice while others dealt with superior knowledge or site inspection. Each of the thirty-seven cases was then located and read to determine if the information reported in the transcript of the case was sufficient for DSCAS to analyze the case. Cases with insufficient information to provide an adequate analysis were discarded. Also, the number of claims and amount of distinction between interrelated claims and facts in cases which concerned multiple claims were appraised. Cases in which facts pertaining to the DSC claim became interwoven with facts pertaining to other claims were discarded also.

At the end of this selection process twelve cases remained containing thirteen different DSC claims. An additional case which had been settled out of court was obtained from the U.S. Bureau of Reclamations files and was added to the list of test cases. Each of the cases was analyzed by DSCAS and one on which DSCAS performed questionably during the initial analysis was rerun. The actual testing of the cases was quite straight-forward. A familiarity was gained with the facts involved in the DSC claim being analyzed prior to beginning the session with DSCAS. Then during the session it was just a matter of selecting the correct response to the questions asked by DSCAS based on the facts of the instant case.

The remaining discussion in this section deals explicitly with the observations made and the insights gained during the testing. This discussion of each case seeks only to reveal accomplishments and problems encountered during the testing process. Transcripts of each of the tests as well as a case summary are included in Appendix C.

11.2.1 Alps Construction Corporation

The Alps Construction Corporation case deals with a DSC claim based on the unexpected occurrence of rock at five locations on the jobsite during contract performance. The analysis performed by DSCAS for this case deals only with the initial encounter with the rock in the vicinity of work pertaining to the ground storage reservoir. Based on the facts and decisions presented in the case DSCAS accurately determined that the contractor should be granted entitlement. The key conclusions drawn for this case were; "contract did contain indications concerning condition," indicating that a Type I condition was encountered and "reasonable inspection is not required for entitlement," indicating that the condition would not have been discovered even if a site inspection had been performed. Also, DSCAS concluded that the "government did have superior knowledge" because additional evidence of the condition was withheld from the bidders. Upon making this conclusion, DSCAS also alerted the user to the possibility of pursuing the claim through government breach of contract. Lastly, the contract contained an exculpatory clause attempting to

deny government liability and responsibility for the actual conditions which differed from those indicated. DSCAS accurately concluded that the "validity of exculpatory clause is in doubt" because the clause was not specific to the DSC and contained language which was unclear.

The only question which could not be answered purely on facts presented in the case was "Was the difference between actual and indicated conditions a material difference?" This question was answered based on the conclusions of the BCA decision. The problem with this question is that it requires too much legal judgment. In the instant case the government asserted that the difference was not material while the contractor contended that the difference was material; therefore, the decision of the BCA was used. Absent the conclusion reached in the BCA decision DSCAS would have had to rely on the legal judgment of the user to resolve this conflict of opinion.

11.2.2 Bureau of Reclamations

The Bureau of Reclamations case dealt with the occurrence of unexpected amounts of water during excavation for a pipeline. This case was settled out of court; therefore, the information used for the DSCAS analysis was obtained from the correspondence between the involved parties. DSCAS accurately predicted that the contracter was entitled to the claim. However, because of the nature in which the facts of this case were obtained, a number of unknowns were encountered. These unknowns all pertained to the information contained by the contract and could have affected DSCAS's performance. Other

than the problem with the question pertaining to material difference mentioned previously, DSCAS was able to collect and accurately interpret the known information in this case.

11.2.3 Continental Drilling Company

In the case involving Continental Drilling Company the contractor encountered excessive cave-ins during the performance of a drilling contract. DSCAS was able to accurately determine that the contractor was entitled to an equitable adjustment through the DSC claim. An important decision reached in this case was that the condition encountered was a Type I condition because certain inferences could be made from the information presented by the contract about the amount of cave-ins to be expected. Also, neither the physical data included in the contract nor a site inspection revealed the actual conditions. Based on these facts DSCAS was able to conclude that neither inspection of the physical data nor inspection of the site were required for entitlement.

Problems encountered during this analysis include those previously mentioned pertaining to material difference (once again the BCA decision was relied on) and a lack of information regarding the notice. However, because of DSCAS's ability to make assumptions in the absence of necessary information, the analysis was able to continue. An additional point to emphasize is the caution used by DSCAS when unknown information is encountered. This cautious

continuation of the analysis became obvious in this case because DSCAS still took the time to check if the government had been prejudiced even though in actuality the contractor may have complied fully with the notice requirements.

11.2.4 DeMauro Construction Company

The DSC encountered in the DeMauro Construction Company case was the unanticipated amount of rock discovered during excavation for a water main. DSCAS performed quite admirably on this case by determining that entitlement would not be allowed. DSCAS arrived at this conclusion for two separate reasons: first, because the government was prejudiced due to the lack of timely notice and, second, because the contractor failed to heed contract indications concerning the existence of the condition.

DSCAS was alerted to check prejudice to the government because the notice was not given before the contractor disturbed the condition or before the contractor performed remedial work. In the actual case notice was not given until after the completion of the work. DSCAS currently has no means for pinpointing this time specifically. The actual analysis of prejudice to government proved to be quite adequate. However, once the occurrence of prejudice hau been concluded, questions twelve and thirteen (see DEMAURO, Appendix C) proved to be rather vague in light of the available information. An additional comment pertaining to the occurence of prejudice is that, in the current configuration of DSCAS's analysis, if the government has been prejudiced, the user is alerted

to the fact and informed whether or not entitlement is still probable based on the answers to questions twelve and thirteen. Then, if entitlement is not still probable, the analysis continues checking for additional reasons for no entitlement. Continuing with the analysis is fine; however, no explanation is given for concluding that no entitlement will be allowed because the government was prejudiced.

In this case the ability to determine reasons for no entitlement other than prejudice alone proved valuable. The second reason for no entitlement occurred when DSCAS discovered that the physical data included in the contract did indicate the nature of existing conditions. Once this second reason for no entitlement was discovered the analysis was stopped. However, had the analysis continued, an additional reason for no entitlement would have been discovered because a reasonable site investigation would have also revealed the existence of the condition.

11.2.5 J. J. Welcome Construction Company, Inc.

The J. J. Welcome Construction Company, Inc. case deals with unexpected muck conditions encountered during clearing and grading of the jobsite. DSCAS accurately predicated that entitlement to an equitable adjustment for the DSC would be allowed. One of the government's contentions in this case was that the contractor had failed to comply fully with the notice requirements for a DSC claim even though the government did receive oral notice of the condition. DSCAS refuted this contention as did the BCA. DSCAS realized that if the government was not prejudiced by improper notice entitlement will not be denied.

Although DSCAS was capable of reaching the correct conclusion, two new problems were encountered during the analysis. First, DSCAS did not adequately handle the information pertaining to site inspection. In this case the contractor was unable to conduct a complete site inspection because the area of the jobsite in which the DSC was encountered was inaccessable, through no act of the government, at the time the site inspection was conducted. DSCAS had no means of grasping this. To get around this problem question twenty-eight (see JJWELCO, Appendix C) was answered with a "No" because a reasonable inspection at the time the inspection was conducted would not have revealed the condition, i.e., a reasonable inspection would not have included inspection of the inaccessable area.

A second point which DSCAS was unable to grasp dealt with the government's contention that the contract failed to indicate that an addtional cost would be incurred because of the DSC. The BCA determined that since the government was cognizant of the condition the contractor need not notify it of additional costs; therefore, DSCAS's inability to handle this concept did not affect the outcome of the claim. Lastly, DSCAS analyzed this case twice to exemplify DSCAS's ability to review a previously analyzed case. The results of both tests, JJWELCO.ORIG and JJWELCO, have been included in Appendix C.

11.2.6 Layne Texas Company

The Layne Texas Company case dealt with two different DSC claims concerning the unexpected occurrence of boulders during

the performance of a drilling contract. The first claim asserted that the condition encountered was a Type I DSC while the second claim asserted that the same condition was a Type II DSC. Both claims were analyzed by DSCAS and both were predicted correctly based on the available information, denying the first and allowing the second. The LAYNE.ORIG and the LAYNE test cases in Appendix C deal with analysis of the Type II DSC while the LAYNE2.ORIG and the LAYNE2 test cases deal with analysis of the Type I DSC assertion.

DSCAS treated the analysis of the Type II DSC claim without any problems except for the determination of material differences. Since the government asserted the condition was not materially different while the contractor contended that the condition was materially different, the decision of the BCA was used to answer DSCAS's question concerning material difference. DSCAS's treatment of the Type I DSC claim was only adequate because the real issue of the condition was whether or not the government misrepresented the conditions one could expect to encounter at the site. Since the contract did contain indications of the conditions at the site, one was required to determine precisely what the indications actually represented. To make this determination legal judgment was required. Therefore, it can be concluded that this question does not adequately handle the underlying facts important to this decision.

11.2.7 C. H. Leavell and Company

The C. H. Leavell and Company analysis was based on four cases which had been decided by the BCA, each concerning the same

DSC claim which was appealed each time. The case centered around an unexpected amount of soil moisture encountered during the performance of a contract for the construction of drilled piers. For this case DSCAS accurately predicted that no entitlement would be allowed through the DSC claim because the contractor had failed to make simple inquiries concerning some of the symbols used in the contract drawings.

Although DSCAS handled this case quite well, a potential problem exists in the analysis of the occurrence of prejudice to the government. DSCAS accurately determined that the government had been prejudiced but was unable to determine whether or not entitlement should be denied because of this prejudice. This problem resulted because the answer to question thirteen was unknown (see LEAVELL.ORIG, Appendix C) causing DSCAS to assume that "additional proof does exist to prove entitlement" which in all probability is incorrect. The determination of whether or not entitlement should be denied because of the prejudice was the central concern of the first two appeals of the case. Since DSCAS was unable to draw any conclusions or, at best, default to allowance of entitlement, this indicates that the process used to determine entitlement once prejudice has been established is inadequate in some situations.

11.2.8 Bernard McMenamy Contractor, Inc.

The DSC in this case occurred when the contractor, Bernard McMenamy Contractor, Inc., encountered large quantities of rock during the performance of a dredging contract. DSCAS accurately

classified this DSC claim as a Type I condition and accurately predicted that entitlement would be allowed. The analysis of this case was straightforward except for the determination of what exactly was meant by a sentence contained in the contract. The determination required some knowledge of the structure of compound sentences. Knowledge of this type is quite understandably not contained by DSCAS. The only other point of difficulty involved determination of whether or not the conditions differed materially from those indicated. This problem has been addressed previously; therefore, the discussion will not be reiterated here. Lastly, this case also involved determination of the validity of an exculpatory clause contained in the contract. DSCAS concluded that the "validity of exculpatory clause is in doubt" which agrees with the findings of the BCA.

11.2.9 Norair Engineering Corporation

The Norair Engineering Corporation encountered a DSC when unexpected groundwater and sewage began flowing into their excavation. This case was very straightforward and DSCAS was able to accurately predict that the contractor was entitled to an equitable adjustment through the DSC claim. No problems deserving of an explanation were encountered during the analysis of this case.

11.2.10 Jack Picoult

The contractor, Jack Picoult, in this case encountered a DSC when the actual conditions encountered while installing duct-work

differed from those indicated by the contract documents. DSCAS accurately predicted that the contractor would be allowed entitlement through the DSC claim. Although DSCAS had no real problems with the analysis of this claim based on the facts and information available, it failed to grasp a concept which the BCA deemed important to the claim. The concept which DSCAS missed is concerned with the extent of the government's acceptance of responsibility for the resolution of the DSC.

In this case the BCA noted that the government did not take the position that responsibility for resolution of the DSC was solely the contractor's responsibility. Instead, the government took control of both the inspection of and the work necessary to resolve the DSC. The BCA used this evidence to determine that the government's contention that the contractor is solely responsible for all work that is required as indicated by the contract documents and site investigation was unjustified. In addition, the BCA concluded that the contract indications were deficient in that the condition was not indicated and the government was unaware of the condition prior to the contractor's notice. DSCAS was only able to refute the government's contention based on these last reasons.

11.2.11 <u>Southwest Engineering Company, Inc.</u>

In this case the contractor, Southwest Engineering Company, Inc., asserted that a DSC had been encountered when an excavation involved unexpected amounts of blasting and drilling. DSCAS handled this case extremely well, concluding that no entitlement would be

allowed through the DSC claim. By running this case twice DSCAS determined two reasons for no entitlement. In the SW-ENGR.ORIG and the SW-ENGR tests DSCAS determined that no entitlement would be allowed because the condition encountered did not differ from the expected conditions which could be established from general knowledge in the industry about site conditions in the area. In the second run, SWENGR2.ORIG and SWENGR2 the analysis was allowed to proceed past the question concerning establishment of expected conditions, by answering "-." The second analysis of this case also concluded that no entitlement was probable, however, this time because the contractor did not make simple inquiries concerning the local site conditions which would have revealed the condition.

Both of the reasons for no entitlement given by DSCAS are valid reasons. However, the BCA determined no entitlement would be allowed because the contractor had imputed knowledge of the condition. The contractor's imputed knowledge resulted from having experienced the same condition during performance of a previous contract at the same jobsite. DSCAS never reached this point in its analysis because other reasons for no entitlement were discovered both times. However, after closely examining the decision process pertaining to the analysis of superior knowledge under a Type II claim or even under a Type I claim, it has been discovered that DSCAS is unable to deal effectively with imputed knowledge. The only type of imputed knowledge DSCAS can currently deal with is knowledge gained by the contractor which has been communicated to him by

the government. Since this is quite obviously not the type of imputed knowledge in question, DSCAS's ability to handle claims dealing with imputed knowledge must be expanded.

11.2.12 Tectonics, Inc.

The contractor in this case, Tectonics, Inc., asserted that a DSC was encountered when a recessed room which was not indicated on the contract drawings was discovered during performance of an electrical contract. DSCAS accurately predicted that the contractor was not entitled to an equitable adjustment through the DSC claim. DSCAS drew this conclusion because the contractor did not perform a reasonable site inspection which would have revealed the existence of the condition. This test case was very straightforward and was analyzed quite easily by DSCAS.

11.3.13 Welch Construction Company, Inc.

The contractor in this case, Welch Construction Company, Inc., asserted that a DSC had been encountered when impervious soil at a jobsite caused excessive ponding of runoff due to unseasonal rains. DSCAS predicted that the contractor was not entitled to an equitable adjustment through the DSC claim, as did the BCA. DSCAS concluded no entitlement because the contractor failed to heed indications concerning the impervious nature of the soil conditions presented in the physical data. The BCA, on the other hand, concluded that no entitlement would be allowed because the rain, which is accurately identified by DSCAS as a non-compensable after-bid condition resulting

from an act-of-God, did not interact with an unknown physical factor at the site. At this point in the analysis DSCAS failed because it asked if the non-compensable after-bid condition had interacted with a physical factor at the site, omitting the word "unknown." The answer to DSCAS's question is obviously "Yes." However, if DSCAS's question had been phrased correctly, corresponding to the BCA's intention, the answer would have been "No," in which case DSCAS would have accurately identified the same reason for no entitlement as the BCA. This problem is rather minor since it merely involves rephrasing the question being asked by DSCAS and its corresponding data base assertions.

11.3 Conclusions Based on DSCAS's Performance

Overall the results from testing DSCAS were extremely encouraging and somewhat better than initially anticipated. In all the cases tested DSCAS was able to accurately determine whether or not entitlement would be allowed even though in a number of cases the reasons for no entitlement were valid but not identical to those determined by the BCA. Most of the problems encountered were centered around the required use of more legal judgment than desired. This section discusses the conclusions which have been drawn from the problems encountered while testing DSCAS.

The most apparent limitation of the analysis capabilities of DSCAS is the amount of legal knowledge and judgment required to correctly answer a number of the questions. In most instances

this need for additional legal discernment results from asking questions at a level which is too high to allow effective selection of the appropriate response, i.e., the questions are not oriented toward determining the underlying facts relative to a given concept, rather they try to determine the concept directly. If you will recall, one of the important functions to be performed by DSCAS was to allow the field personnel to analyze a claim based on his technical knowledge. To achieve this goal the current version of DSCAS must be further refined to reduce the amount of legal discernment required.

A related limitation encountered when using the records of previous cases was the determination of whose argument was correct, the contractor's or the government's. The test cases did not contain enough information from the viewpoint of any one party to allow an analysis to be built solely around one party's arguments. Therefore, all the information presented in the case had to be assimilated to allow a complete analysis. However, if DSCAS were to be refined to the degree suggested in the above paragraph DSCAS should theoretically be able to do the assimilation internally simply by asking enough questions pertaining to a given concept to sort out the arguments and determine whose position is correct. Once DSCAS has been developed to this degree of refinement, legal judgment problems, such as that encountered when answering the question pertaining to material difference, would be resolved simply by having DSCAS ask questions which lead the user through the underlying facts which are used to determine the concept.

Another limitation of the current configuration of DSCAS is that it is unable to determine more than one reason for no entitlement per session. Even though DSCAS does not stop the analysis when no entitlement will probably result because the government was prejudiced the user cannot currently obtain an explanation of why this conclusion was drawn. A number of the test cases which were run would have produced more than one reason for no entitlement had DSCAS allowed the analysis to continue. A good example of this is the case involving Southwest Engineering Company, Inc., which would have caused at least three reasons for no entitlement to be concluded by DSCAS.

The limitations specifically due to the current logic and wording of questions were extremely rare. In fact, only three became evident during this testing. These three include: the treatment of imputed knowledge, the analysis of a site inspection when it has been hindered by events beyond the control of either party and treatment of notice compliance when notice is given after performance of the work or completion of the job. Each of these three limitations can be alleviated simply by expanding DSCAS's assortment of questions pertaining to these concepts.

Finally, two additional conclusions which can be drawn are: 1) DSCAS quite adequately demonstrates that much potential exists for further application of AI to claim analysis and other areas of contract management, and 2) DSCAS has demonstrated that it possesses the potential to accurately analyze the DSC claim. Therefore, one can conclude that DSCAS warrants further work to continue the

development of the corrections and additional features recommended by this thesis. Although the fourteen claims used to test DSCAS quite obviously did not adequately test all of DSCAS's analysis capabilities they did demonstrate that an AI system could be developed which is capable of analyzing the DSC claim. The relative accuracy of the analysis and the minimal amount of problems discovered during testing strongly support both of the above conclusions.

Before making recommendations for further directions of this study a few observations and suggestions which pertain to the testing procedure have been included. Extensive testing of DSCAS is strongly recommended prior to additional work to pinpoint problem areas and other areas within the system most deserving of additional development. This testing should be carried out by persons who are unfamiliar with DSCAS's internal logic structure working closely with a person who is quite familiar with DSCAS's configuration. By using this method of testing, problems with subconscious interpretation of a question as it relates to DSCAS's overall logic structure would not bias the response to a question. This method of testing would also lend itself toward refinement of the wording of questions and provide a more realistic setting for the tests to be performed in. In addition, since DSCAS is developed for use at a jobsite by the government or owner's field personnel, tests should be run which attempt to represent only the facts and arguments of one of the parties in the contract.

11.4 Recommendations for Further Development of DSCAS

DSCAS has demonstrated an ability to analyze the DSC claim. However, to reach the goal of actual field application and use, many refinements and additional features are necessary. This final section makes recommendations for additional development of the DSCAS program. The first group of recommendations pertain to the capabilities currently possessed by DSCAS which should be expanded and refined. The last group of recommendations suggest additional features which should be added to enhance DSCAS's capabilities. Also, since the previous section presented some recommendations as part of the discussion of the conclusions drawn from the test results those recommendations will not be reiterated in this discussion.

In the DSCAS program the feature needing the most immediate attention is the use of the data bases. As was mentioned previously, the six data bases which DSCAS currently relies on are too cumbersome to manipulate effectively within the current control structure. The number of data bases should be reduced. As part of this alteration the structure of the propositions asserted to the data bases should be closely scrutinized. Currently these data base sentences are very readable and look quite impressive when a data base is displayed. However, their underlying effect is to reduce the overall implementation of some of ROSIE's more powerful capabilities, such as data base searches for an element matching a given description. The data base sentences should be broken down into more primitive forms, thus allowing full use of ROSIE's potential. These two changes may

seem quite simple at first but actually they involve a complete overhaul of DSCAS's control structure, so beware.

Another feature of DSCAS waiting to be developed further is the explanation capabilities. The rulesets used to explain no entitlement are currently structured so that capabilities for explaining the logic behind asking a certain question can be added simply by adding a few rulesets and rearranging the location of rules which add assertions to the REPORTS data base. The current configuration requires that all conclusions be drawn in the CONCLUDE files. However, at present only those conclusions which indicate no entitlement are drawn there. Therefore, all rules which assert conclusions into the REPORTS data base must be moved into a "To draw-(logic module name)-conclusions" ruleset in the appropriate CONCLUDE file. Also, the necessary commands to invoke each of these rulesets must be added to the COMPLETE ANALYSIS module. After these changes have been made, additional rulesets must be constructed to assist in the process of correctly locating the ruleset with which each question is associated. This can be done with a generator which produces some portion or all of the ruleset header for the ruleset whose conclusion is associated with the question being explained. The generator ruleset should produce its information based on the module currently being checked and the identification code currently assigned to each question for this purpose.

Currently DSCAS relies quite heavily on embedded procedural knowledge for the control of all aspects of the analysis. As DSCAS's

capabilities are expanded and the amount of legal discernment required by the user is reduced, care must be taken to keep as much of the new knowledge as possible from becoming embedded knowledge. While embedded knowledge is quite useful and effective in the environment provided by ROSIE this type of knowledge is quite inflexible. As the system becomes larger and more complex the amount of flexibility built into the system will become crucial to any modifications which need to be made.

Some additional recommendations include the development of a means to flag assumptions made by the system which have a major impact on the claim and the development of a module which would allow the user to change answers to questions which have been answered previously. In addition, the necessary rulesets should be added to allow positive verification of dates and to provide the ability for comparing dates to determine the order of events. Lastly, an important change to the analysis process would be to revise the control structure so that the user would be notified if a reason for no entitlement was discovered and then let the analysis continue. Then, at the end of the analysis, each reason and the corresponding explanation would be displayed for the user. If this capability were developed, the process of locating and constructing the explanation could be carried out simultaneously with the remaining analysis as a background process.

Finally, as part of the recommendations for further work on DSCAS, the development of some powerful capabilities to enhance

the analysis process is suggested. The next most obvious enhancement of DSCAS's capabilities would be to develop the modules necessary to recommend a settlement for the parties involved in the case once the analysis is complete. Another feature which could be used to enhance the analysis capabilities of DSCAS is the implementation of a probability scheme which would assign a degree of certainty to each of the major conclusions reached during the case. Two other intriguing capabilities which would greatly assist DSCAS's analysis process would be the ability to learn from past cases, allowing DSCAS to monitor and refine its own logic structure, and the ability to apply professional judgment when necessary. Lastly, the ultimate goal would be to incorporate DSCAS into a complete system capable of analyzing any type of construction claim.

BIBLIOGRAPHY

Barr, A. and Feigenbaum, Edward A., eds., 1981. The Handbook of Artificial Intelligence. Vol. I. Los Altos: William-Kaufmann, Inc.

Barr, A. and Feigenbaum, E.A., 1982. The Handbook of Artificial Intelligence. Vol. II. Los Altos, Calif.: William-Kaufmann, Inc.

Berliner, H., 1979. "The B* Tree Search Algorithm: A Best-First Proof Procedure." Artificial Intelligence, 12, 23-40.

Bernhard, R., Assoc. Ed., 1980. "An Electronic Advisor/Companion." IEEE Spectrum, 17 (Sept.), 39-43.

Bing, J., 1980. "Legal Norms, Discretionary Rules and Computer Programs" in Computer Science and Law, Ed. B. Niblett. Cambridge, England: Cambridge University Press.

Bledsoe, W.W., 1977. "Non-Resolution Theorem Proving." Artificial Intelligence, 9, 1-35.

Buchanan, B.G. and Headrick, T.E., 1970. "Some Speculation About Artificial Intelligence and Legal Reasoning" Stanford Law Review, Vol. 23. Nov. pp. 40-62.

Campbell, A.N., et al., 1982. "Recognition of a Hidden Mineral Deposit by an Artificial Intelligence Program." Science, 217 (September 1982), pp. 927-929.

Cibinic, J., Jr., Nash, R.C., Jr., Ed. 1981. Administration of Government Contracts. George Washington University.

Cohen, P.R. and Feigenbaum, E.A., eds., 1982. The Handbook of Artificial Intelligence. Vol. III. Los Altos, Calif.: William-Kaufmann, Inc.

Cook, S., et al., 1981. "The Application of Artificial Intelligence to Law: A Survey of Six Current Projects." In Proceedings of the 1981 National Computer Conference. Eds., A. Orden. Arlington, VA: AFIPS Press, pp. 689-696.

Currie, et al., 1971. "Differing Site [changed] Conditions." In Briefing Papers. Wash. D.C., Federal Publications Inc.

Davis, R., 1979. "Interactive Transfer of Expertise: Acquisition of New Inference Rules." Artificial Intelligence, 12, 121-157.

Davis, R., 1980. "Meta-Rules: Reasoning About Control." Artificial Intelligence, 15, 179-222.

Davis, R., Buchanan, B., and Shortliffe, E., 1977. "Production Rules As a Representation for a Knowledge-Based Consultation Program." Artificial Intelligence, 8, 15-45.

Davis, R., and King, J., 1977. "An Overview of Production Systems." Machine Intelligence, 8, 300-332.

Doyle, J., 1979. "A Glimpse of Truth Maintenance." In Artificial Intelligence: An MIT Perspective. Vol. I. Eds. Patrick H. Winston and Richard H. Brown. Cambridge, Mass.: The MIT Press, pp. 117-135.

Duda, R.O., and Gaschnig, J.G., 1981. "Knowledge-Based Expert Systems Come of Age." BYTE, 6 (Sept.), 238-242, 246-248, 254-255, 258-281.

Edelson, E., 1982. "Expert Systems - Computers That Think Like People." Popular Science, Sept. 1982, pp. 48-58.

Fahlman, S.E., and Steele, G.L., 1982. Tutorial on AI Programming Technology: Languages and Machines. Menlo Park, CA: American Association for Artificial Intelligence.

Fain, J., et al., 1981. The Rosie Language Reference Manual. N-1647-ARPA. Santa Monica, CA: The Rand Corporation.

Fain, J., et al., 1982. Programming in ROSIE: An Introduction by Means of Examples. N-1646-ARPA. Santa Monica, CA: The Rand Corporation.

Feigenbaum, E.A., 1978. "The Art of Artificial Intelligence - Themes and Case Studies of Knowledge Engineering." In Proceedings of the 1978 National Computer Conference. Eds. S.P. Ghosh and L.Y. Liu. Montvale, NJ: AFIPS Press, pp. 227-240.

Fiedler, H., 1980. "Functional Relations Between Legal Regulations and Software" In Computer Science and Law, Ed. B. Niblett. Cambridge, England: Cambridge University Press.

Foster, J.M., 1977. "Programming Language Design for the Representation of Knowledge." In Machine Intelligence. Eds. E.W. Elcock and D. Michie. New York: Halsted Press. Vol. 8, pp. 209-222.

Gelernter, H., 1963. "Realization of a Geometry-Theorem Proving Machine." In Computers and Thought. Eds. Edward A. Feigenbaum and Julian Feldman. New York: McGraw-Hill, pp. 134-152.

Glorioso, R.M., and Colon Osorio, F.C., 1980. Engineering Intelligent Systems Concepts, Theory and Application. Bedford, Mass.: Digital Equipment Corp.

Hayes-Roth, F., Waterman, D.A. and Lenat, D.B., Ed., 1983. Building Expert Systems, Reading, Mass., Addison-Wesley Publishing Co., Inc.

Hayes-Roth, F., et al., 1981. Rationale and Motivation for Rosie. N-1648-ARPA. Santa Monica, CA: The Rand Corporation.

Henschen, L.J., 1976. "Tutorial on Resolution." IEEE Transactions on Computers, Vol. C-25, No. 8, August 1976, pp. 769-772.

Hewitt, C., 1972. Description and Theoretical Analyses (Using Schemata) of PLANNER, A Language for Proving Theorems and Manipulating Models in a Robot. Rep. No. TR-258, AI Laboratory, MIT.

Kipps, J., 1983. "ROSIE 2.3 - Update Document." Santa Monica, CA: The Rand Corporation.

Knuth, D.E., and Moore, R.W., 1975. "An Analysis of Alpha-Beta Pruning." Artificial Intelligence, 6, 293-326.

Kuipers, B., 1975. "A Frame for Frames: Representing Knowledge for Recognition." In Representation and Understanding: Studies in Cognitive Science. Eds. D.G. Borrow and A. Collins. New York: Academic Press, 151-184.

Lawlor, R.C., 1980. "Computer Analysis of Judicial Decisions." In Computer Science and Law, Ed. B. Niblett. Cambridge, England: Cambridge University Press.

Lawlor, R.C., 1981. "Analysis and Prediction of Judicial Decisions." In Computers and the Law; An Introductory Handbook. 3rd ed., Ed. R.P. Bigelow. Chicago: Commerce Clearing House, pp. 81-84.

LeFaivre, R.A., 1976. "Procedural Representation in a Fuzzy Problem-Solving System." In Proceedings of the 1976 National Computer Conference. Ed. S. Winkler. Montvale, NJ: AFIPS Press, pp. 1069-1074.

Lenat, Douglas B., 1978. "The Ubiquity of Discovery." In Proceedings of the 1978 National Computer Conference. Eds. S.P. Ghosh and L.Y. Liu. Montvale, NJ: AFIPS Press, pp. 241-256.

Lenat, D.B., 1982. "The Nature of Heuristics." Artificial Intelligence 19, 189-249.

Levesque, H.J., 1980. "Incompleteness in Knowledge Bases." SIGART Newsletter, 74 (Jan.), 150-152.

Maurer, W.P., 1972. The Programmer's Introduction to LISP. New York: American Elsevier.

McCarty, L.T., 1977. "Reflections on TAXMAN: An Experiment in Artificial Intelligence and Legal Reasoning." In Harvard Law Review, Vol. 90, No. 5, March pp. 837-893.

McCarty, L.T., 1980a. "Some Requirements for a Computer-Based Legal Consultant." In Proceedings of the First Annual National Conference on Artificial Intelligence. Stanford, Calif.: n.p., pp. 298-300.

McCarty, L.T., 1980b. "The TAXMAN Project: Towards a Cognitive Theory of Legal Argument," In Computer Science and Law, Ed. B.Niblett. Cambridge, England: Cambridge University Press.

McDermott, D., and Doyle, J., 1980. "Non- Monotonic Logic I." Artificial Intelligence, 13, 41-72.

Meldman, J.A., 1977. "A Structural Model for Computer Aided Legal Analysis," In Rutgers Journal of Computers and the Law. Vol. 6, No.1,pp.27-72.

Meltzer, B., 1970. "Semantics of Induction and the Possibility of Complete Systems of Inductive Inference." Artificial Intelligence, 1, 3, 189-192.

Michalski, R.S., and Chilausky, R.L., 1980. "Knowledge Acquisition by Encoding Expert Rules Versus Computer Induction from Examples: A Case Study Involving Soybean Pathology." International Journal of Man-Machine Studies, 12, 63-87.

Michie, D., ed., 1979. Expert Systems in the Micro Electronic Age. Edinburgh: Edinburgh University Press.

Mylopoulos, J., 1980. "An Overview of Knowledge Representation." SIGART Newsletter, 74, (Jan.), 5-12.

Niblett, B., 1980. "Computer Science and Law: An Introductory Discussion," In Computer Science and Law, Ed. B. Niblett. Cambridge, England: Cambridge University Press.

Niblett, B., 1981. "The Computer as a Consultation Machine" In Computers and the Law; An Introduction Handbook, 34d. ed. Ed. R.P. Bigelow. Chicago: Commerce Clearing House, pp. 90-97.

Nilsson, N.J., 1971. Problem Solving Methods in Artificial Intelligence. New York: McGraw-Hill.

Nilsson, N.J., 1980. Principles of Artificial Intelligence. Palo Alto, Calif.: TIOGA.

Popp, W.G., and Schlink, B., 1975. "JUDITH, A Computer Program to Advise Lawyers in Reasoning a Case," In Jurimetrics Journal, Vol. 15, No. 4, pp. 303-314.

Quinlan, J.R., 1979. "Discovering Rules by Induction from Large Collections of Examples." In Expert Systems in the Micro Electronic Age. Ed. Donald Micnie. Trowbridge, Wiltshire, Great Britain: Edinburgh University Press, 168-201.

Quinlan, J.R., 1980. "An Introduction to Knowledge-Based Expert Systems." The Australian Computer Journal, 12, No. 2, May, pp. 56-62.

Raphael, B., 1976. The Thinking Computer. San Francisco: W.H. Freeman and Company.

Samuel, A.L., 1963. "Some Studies in Machine Learning Using the Game of Checkers." In Computers and Thought. Eds. Edward A. Feigenbaum and Julian Feldman. New York: McGraw-Hill, pp. 71-105.

Simon, M.S., 1979. Construction Contracts and Claims. New York: McGraw-Hill, Inc.

Simon, M.S., 1982. Construction Law Claims and Liability. Butler, New Jersey: Arlyse Enterprises.

Slagle, J.R., 1971. "Artificial Intelligence: The Heuristic Approach." New York: McGraw-Hill.

Stefik, M., et al., 1982. "The Organization of Expert Systems, A Tutorial." Artificial Intelligence, 18, 135-173.

Stone, J., 1964. "Man and Machine in the Search for Justice." Stanford Law Review, Vol. 16, May, pp. 515-560.

Uhr, L., 1973. Pattern Recognition, Learning and Thought Computer Programmed Models of Higher Mental Processes. Englewood Cliffs, N.J.: Prentice Hall.

Vanden Bosche, R.P., 1981. "A Construction Claims Dictionary," CMC Publishing, Camden, N.J.

Waltz, D.L., 1982. "Artificial Intelligence." Scientific American, 247, (Oct.), 118-133.

Waterman, D.A., and Peterson, M.A., 1981. Models of Legal Decision-Making. R-2717-ICJ, Santa Monica, CA: The Rand Corporation.

Williams, G., Sr. ed., 1981a. "Tree Searching Part I: Basic Techniques." BYTE, 6, (Sept.), 72-106.

Williams, G., Sr. ed., 1981b. "Tree Searching Part 2: Heuristic Techniques." BYTE, 6, (Oct.), 195-210.

Winograd, T., 1980. "Extended Inference Modes in Reasoning by Computer Systems." Artificial Intelligence, 13, 5-26.

Winston, P.H., 1977. Artificial Intelligence. Reading, Mass.: Addison-Wesley.

Winston, P.H., and Horn, B.K.P., 1981. LISP. Reading, Mass.: Addison-Wesley.

APPENDIX A

DSCAS LOGIC MODULES

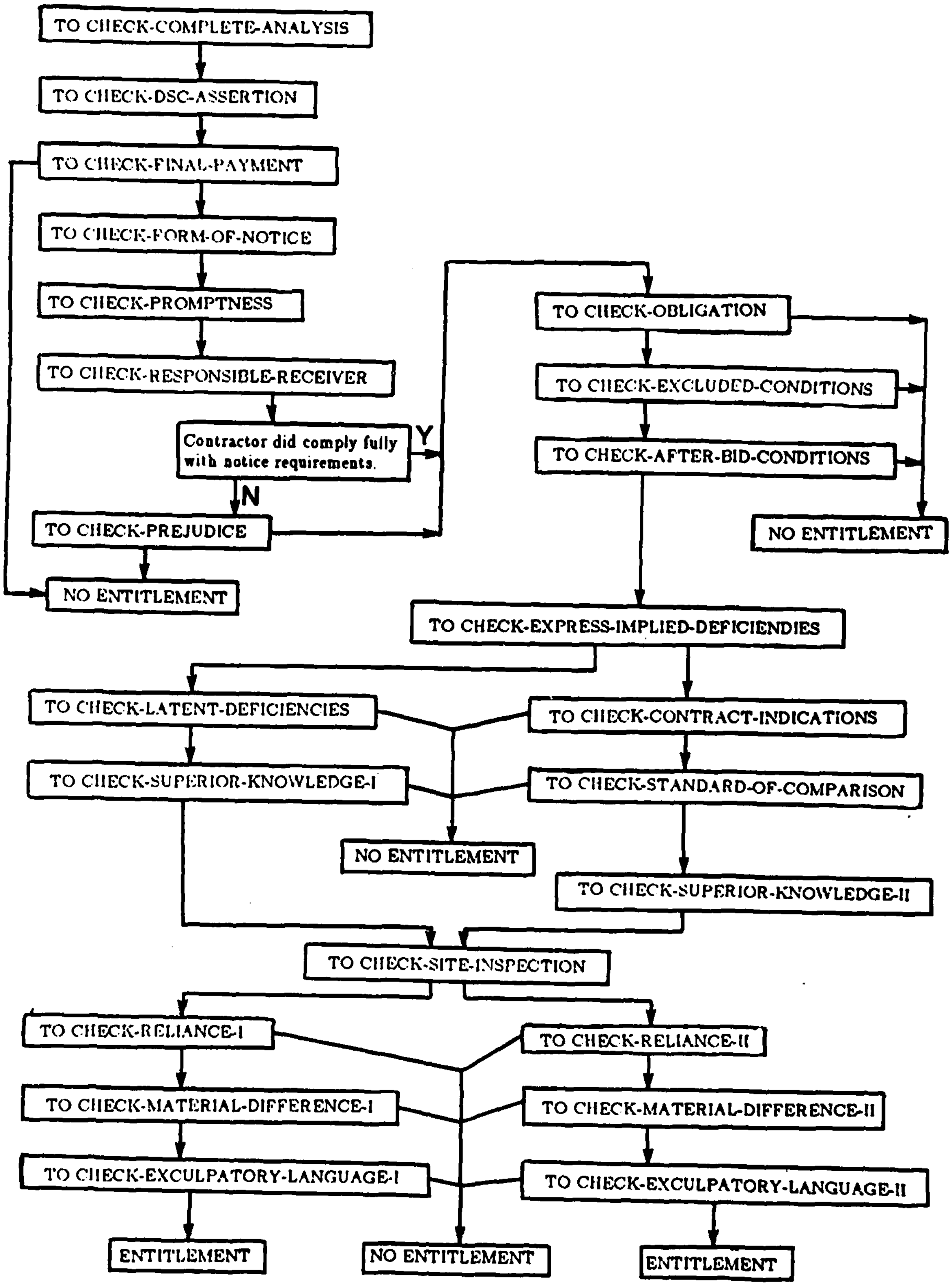


Figure A.1 Complete Analysis

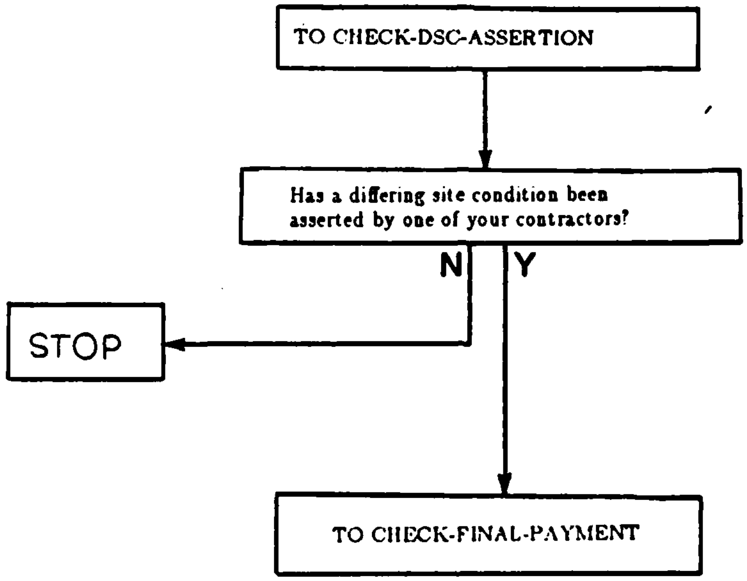


Figure A.2 DSC Assertion

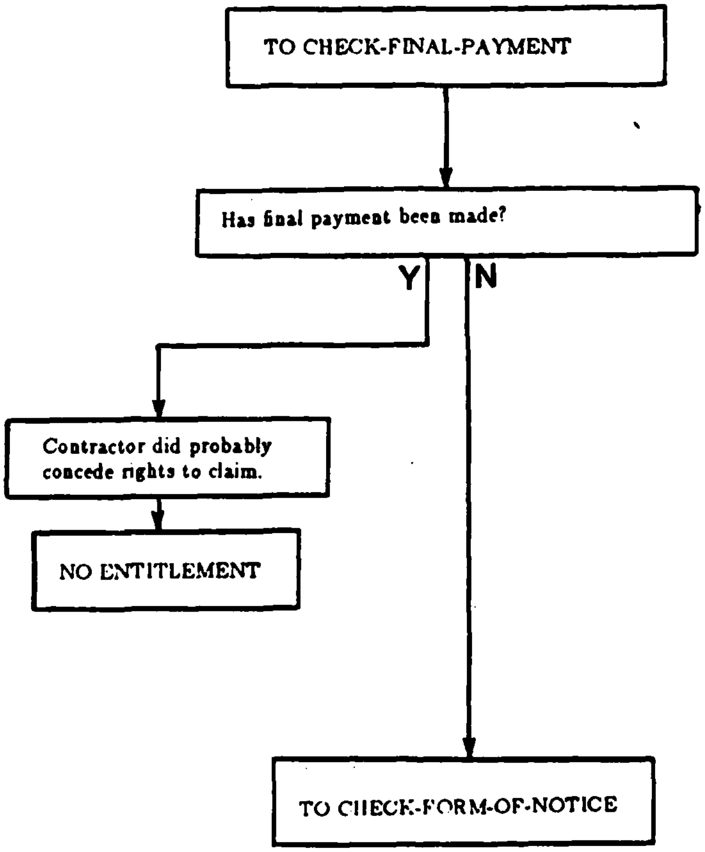


Figure A.3 Final Payment

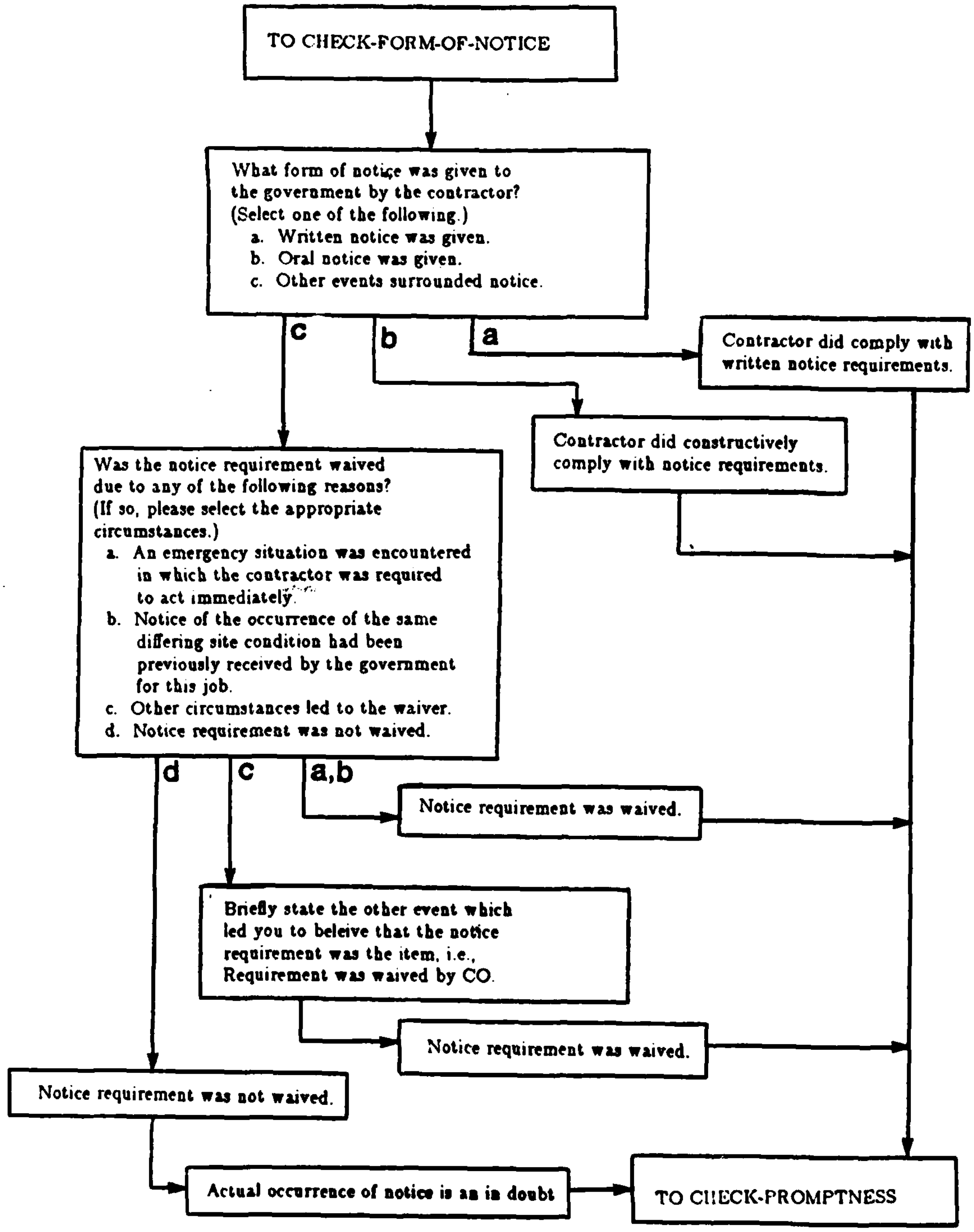


Figure A.4 Notice Form

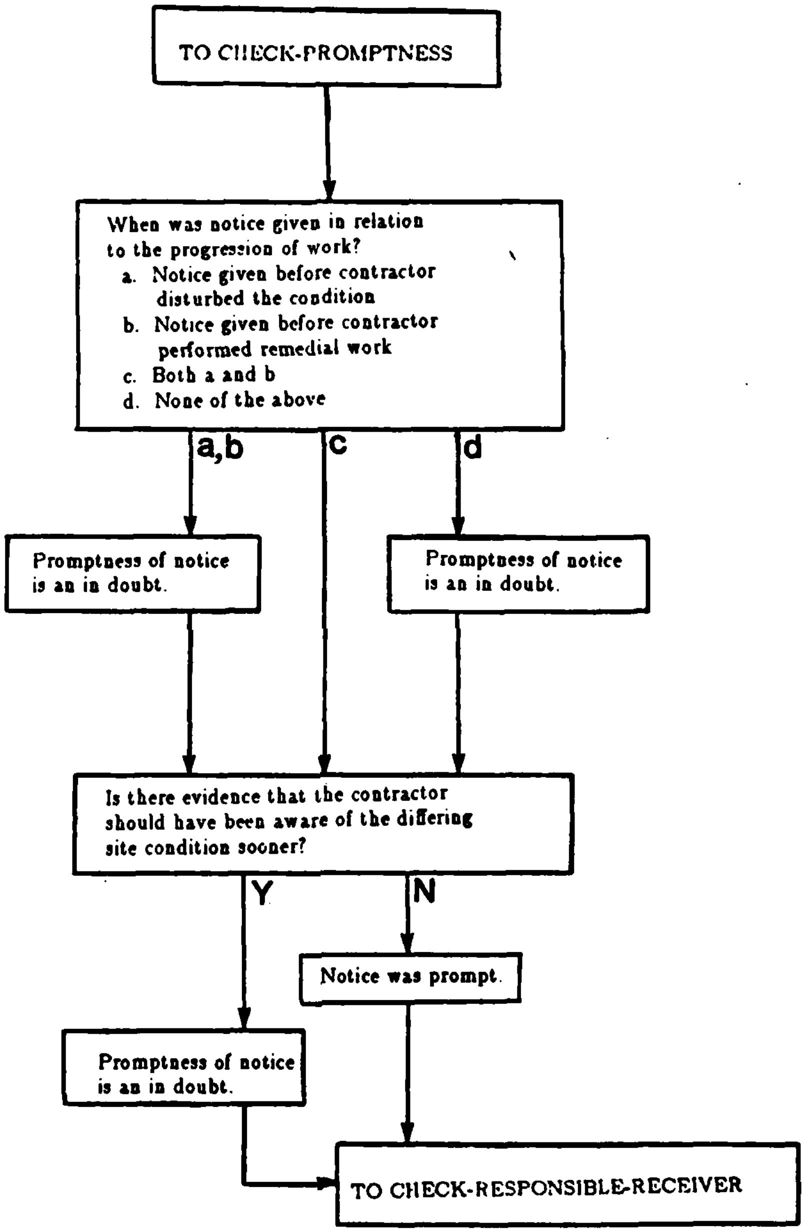


Figure A.5 Notice Promptness

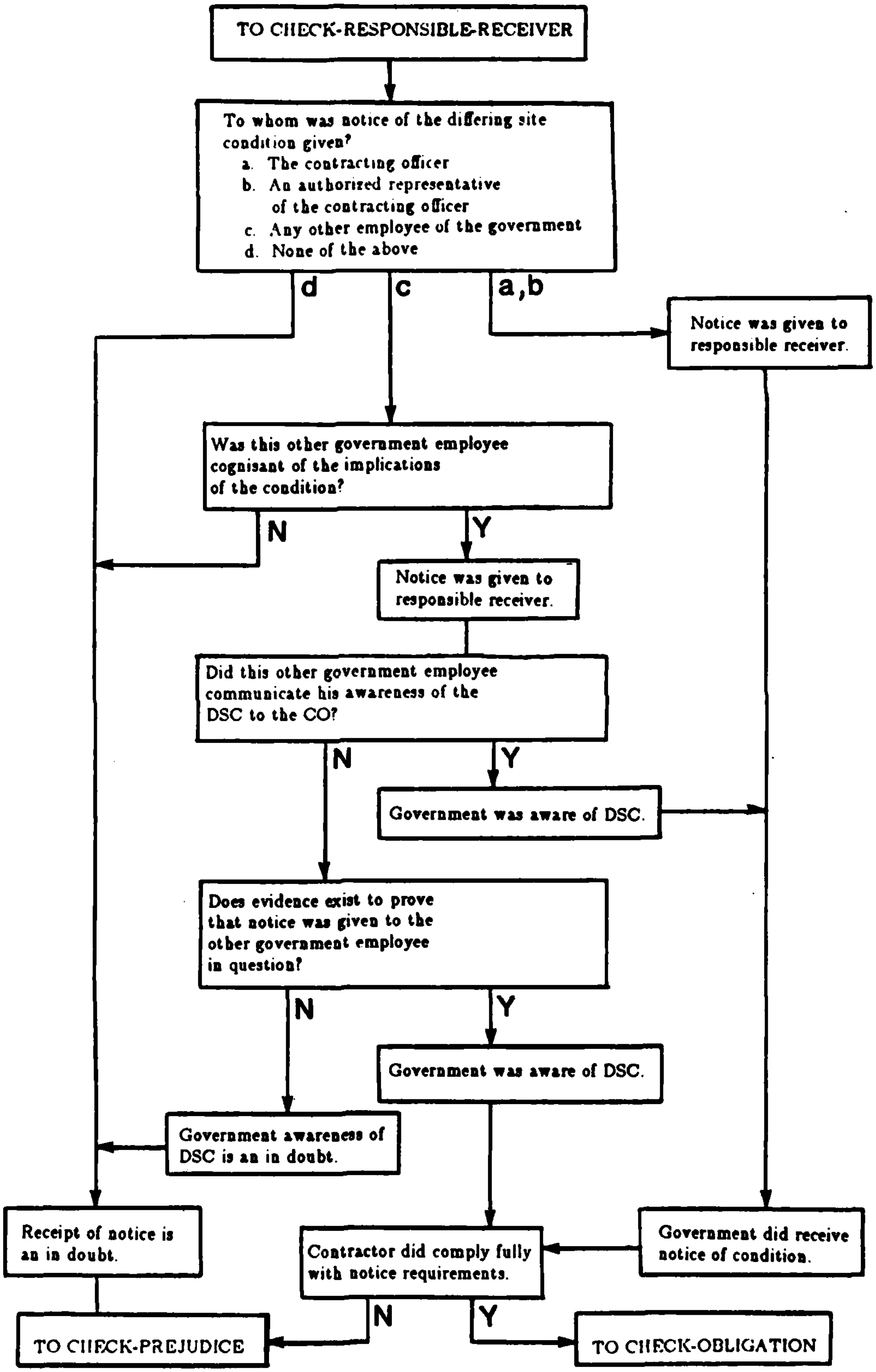


Figure A.6 Responsible Receiver

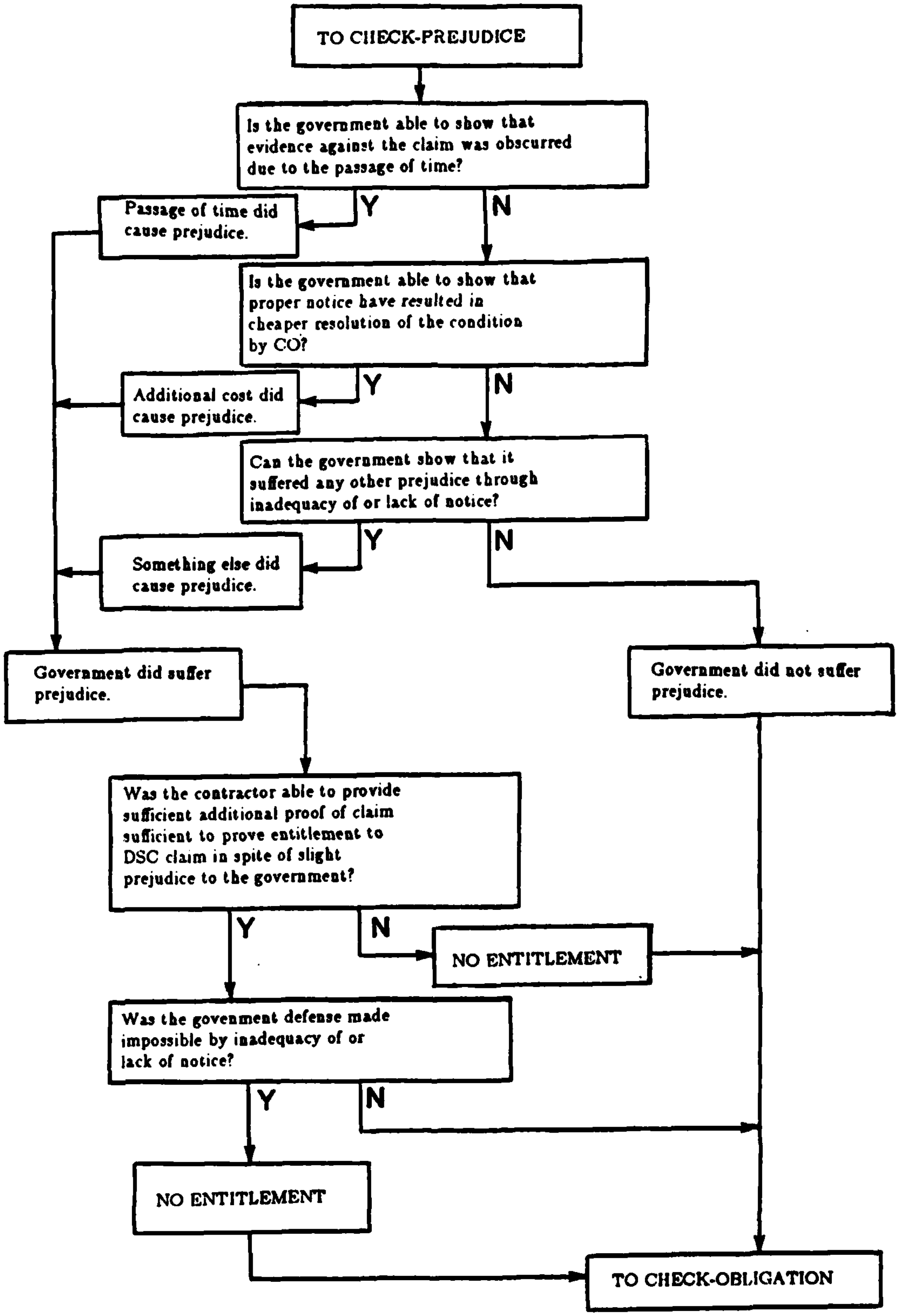


Figure A.7 Government Prejudiced

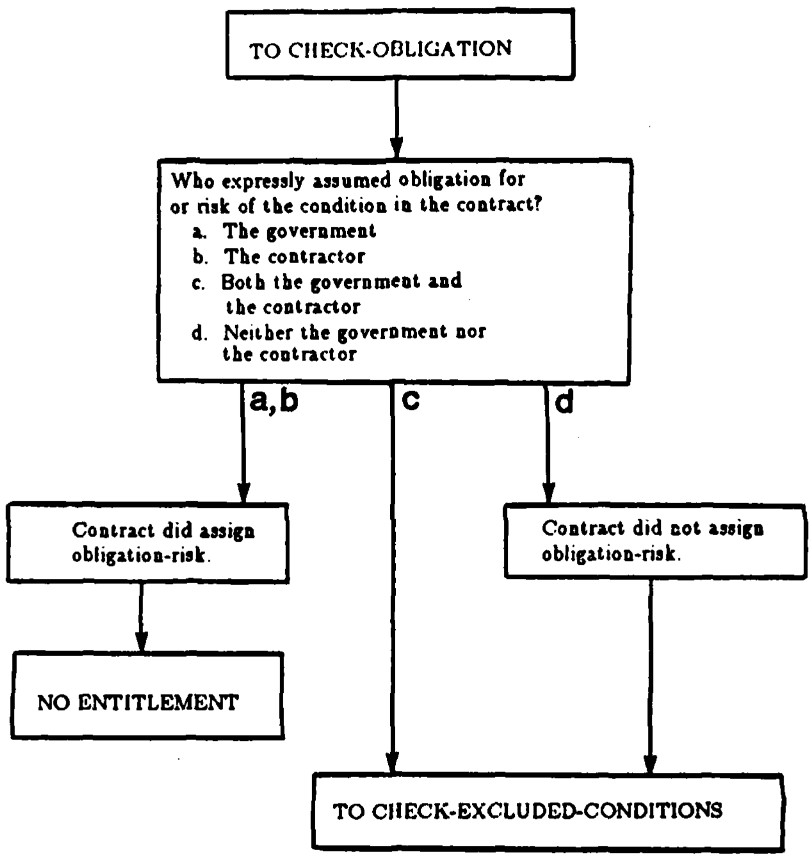


Figure A.8 Contract Obligation

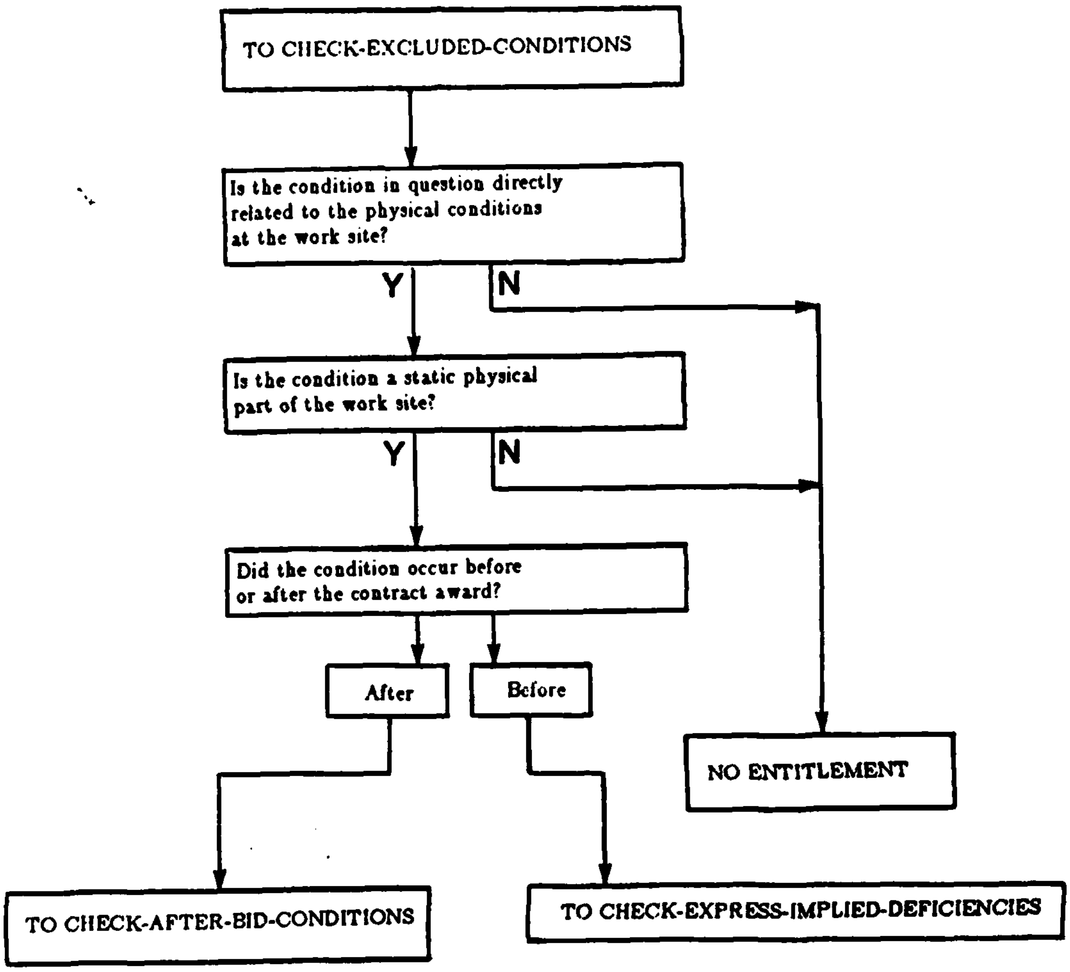


Figure A.9 Excluded Conditions

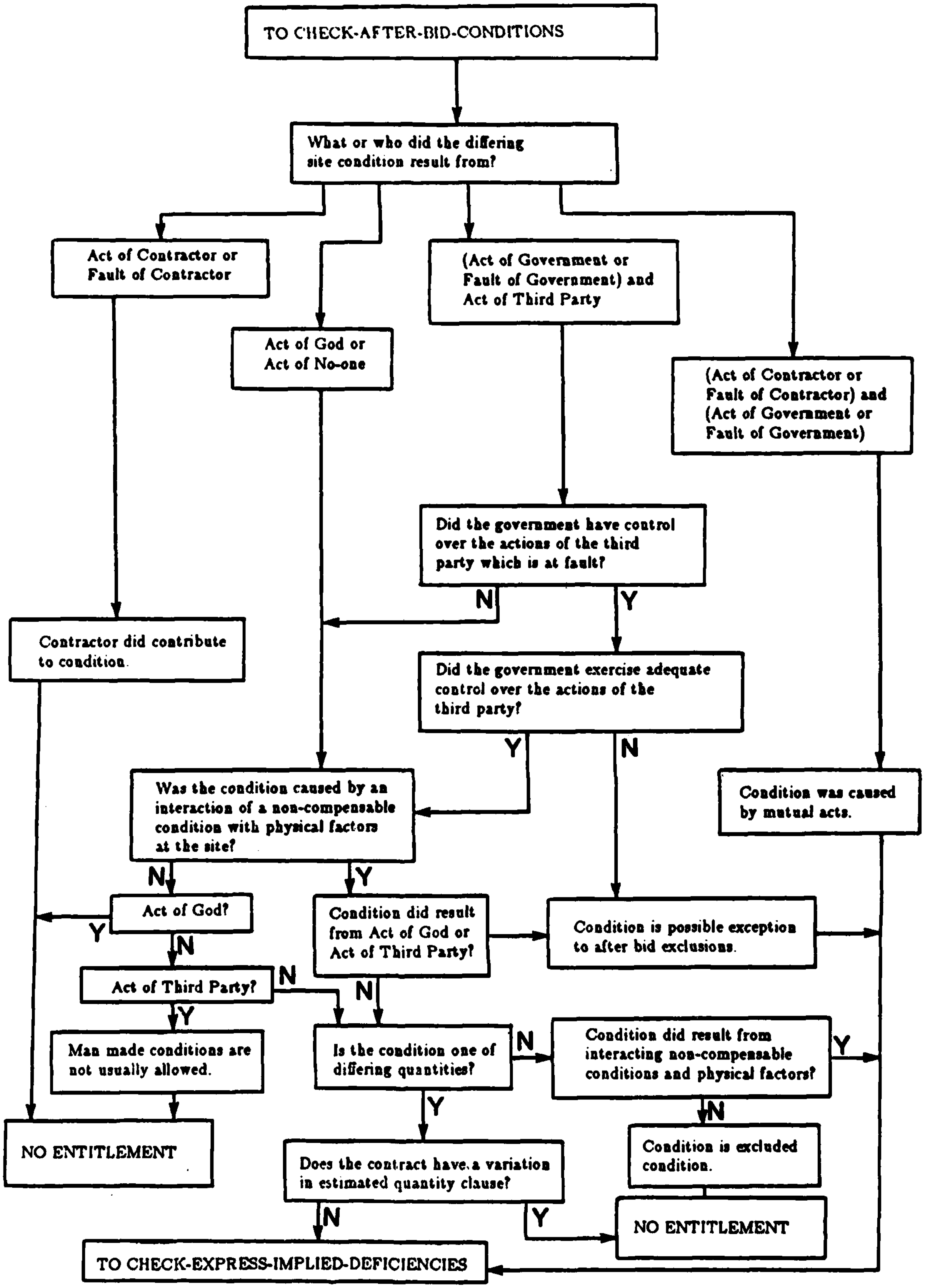


Figure A.10 After-Bid Conditions

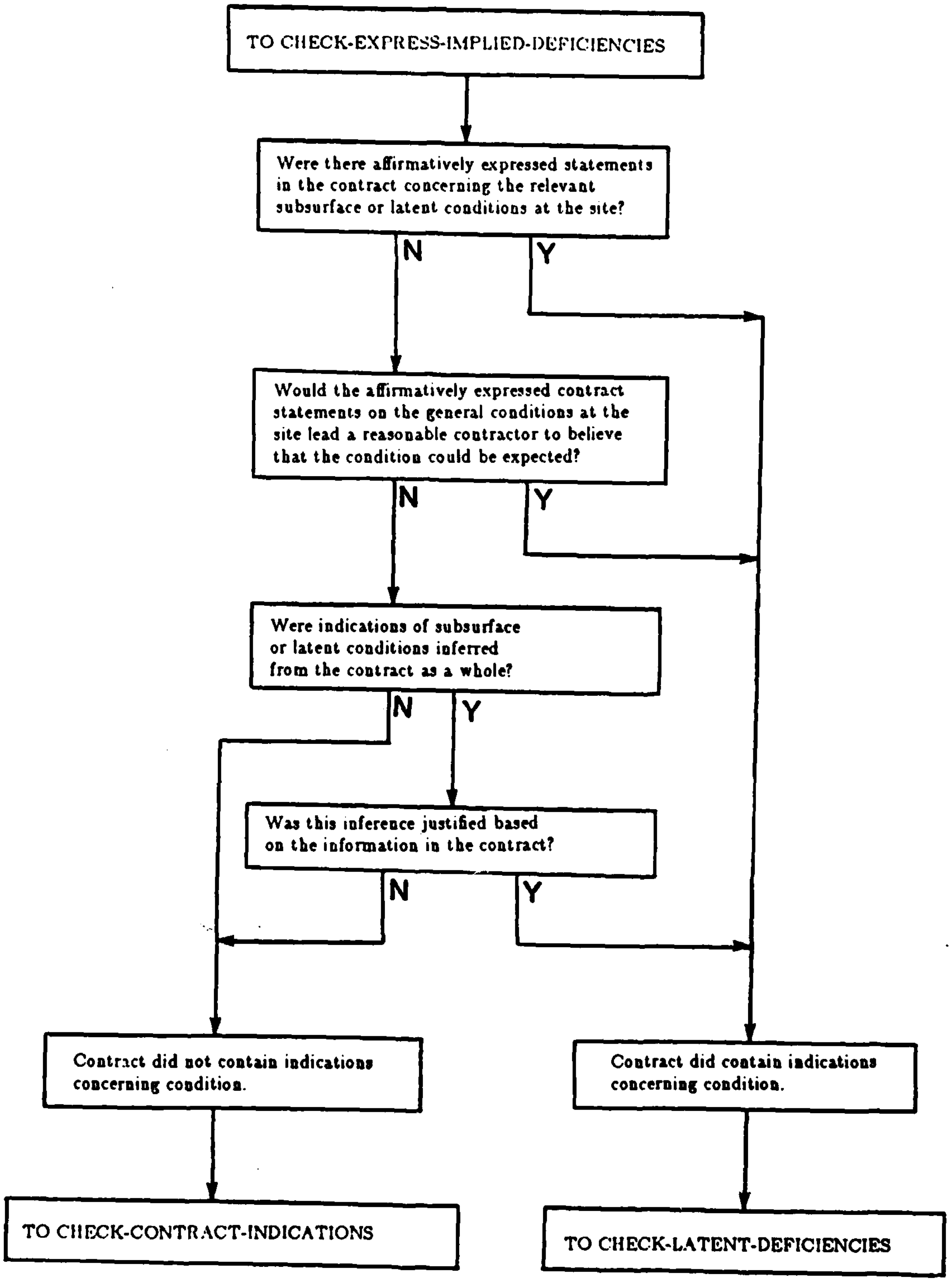


Figure A.11 Express-Implied Conditions

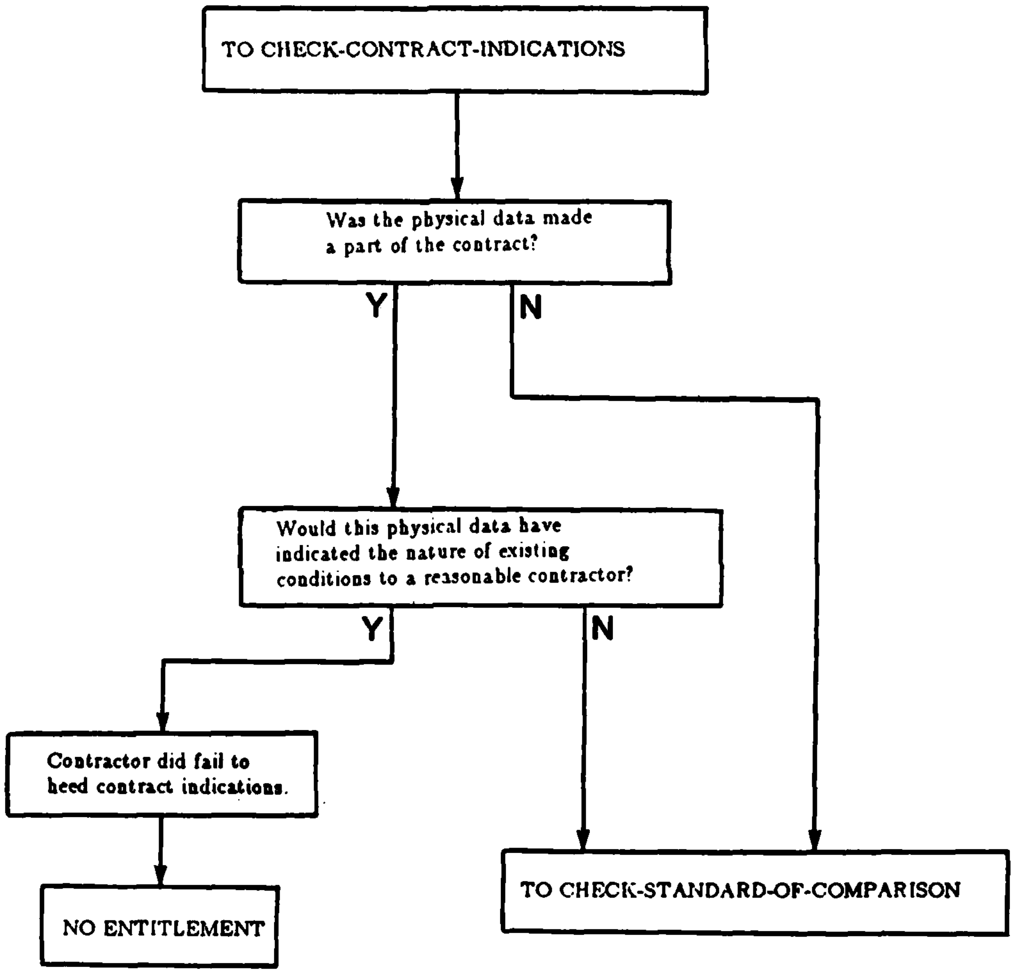


Figure A.12 Contract Indications

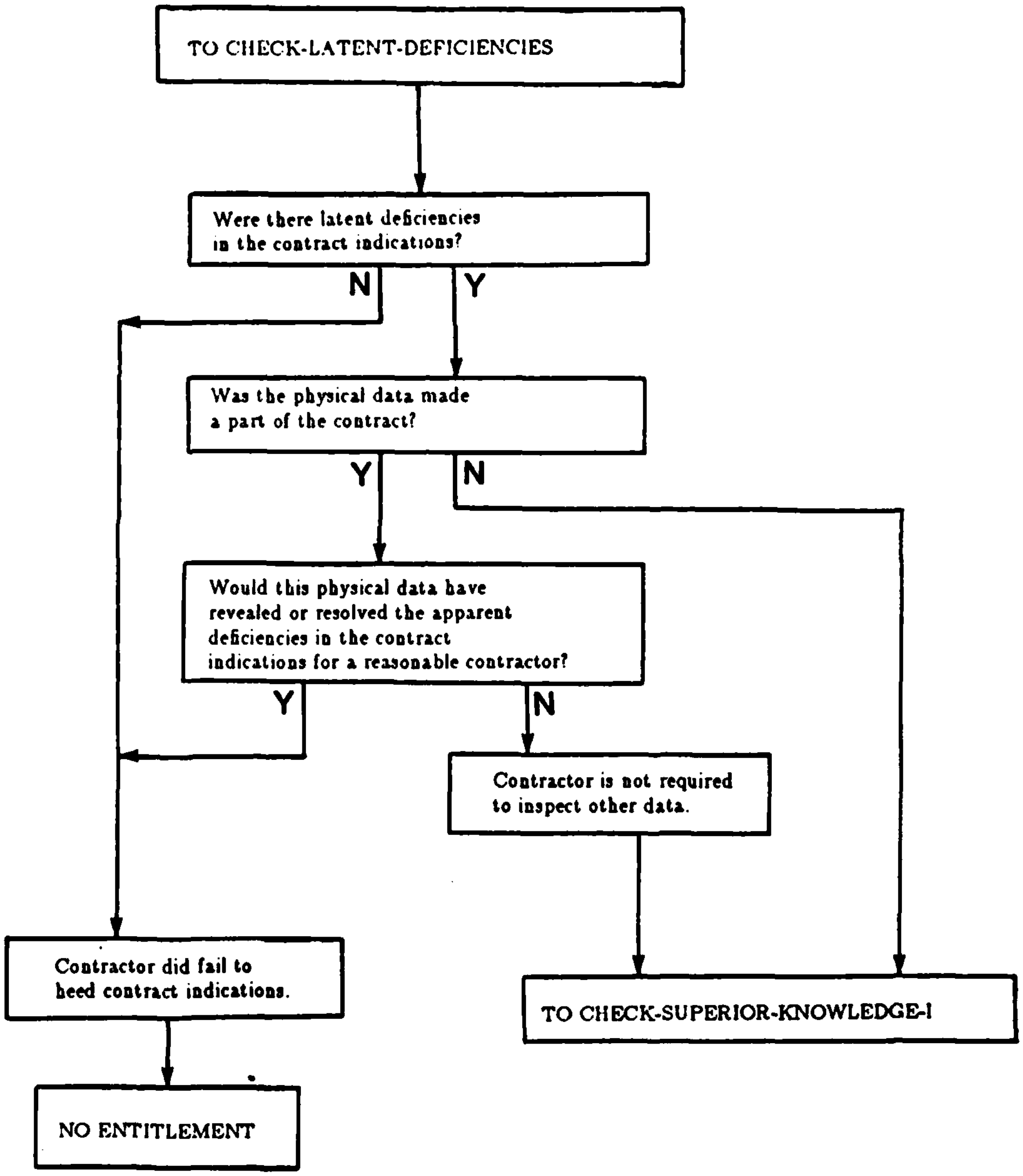


Figure A.13 Latent Deficiencies

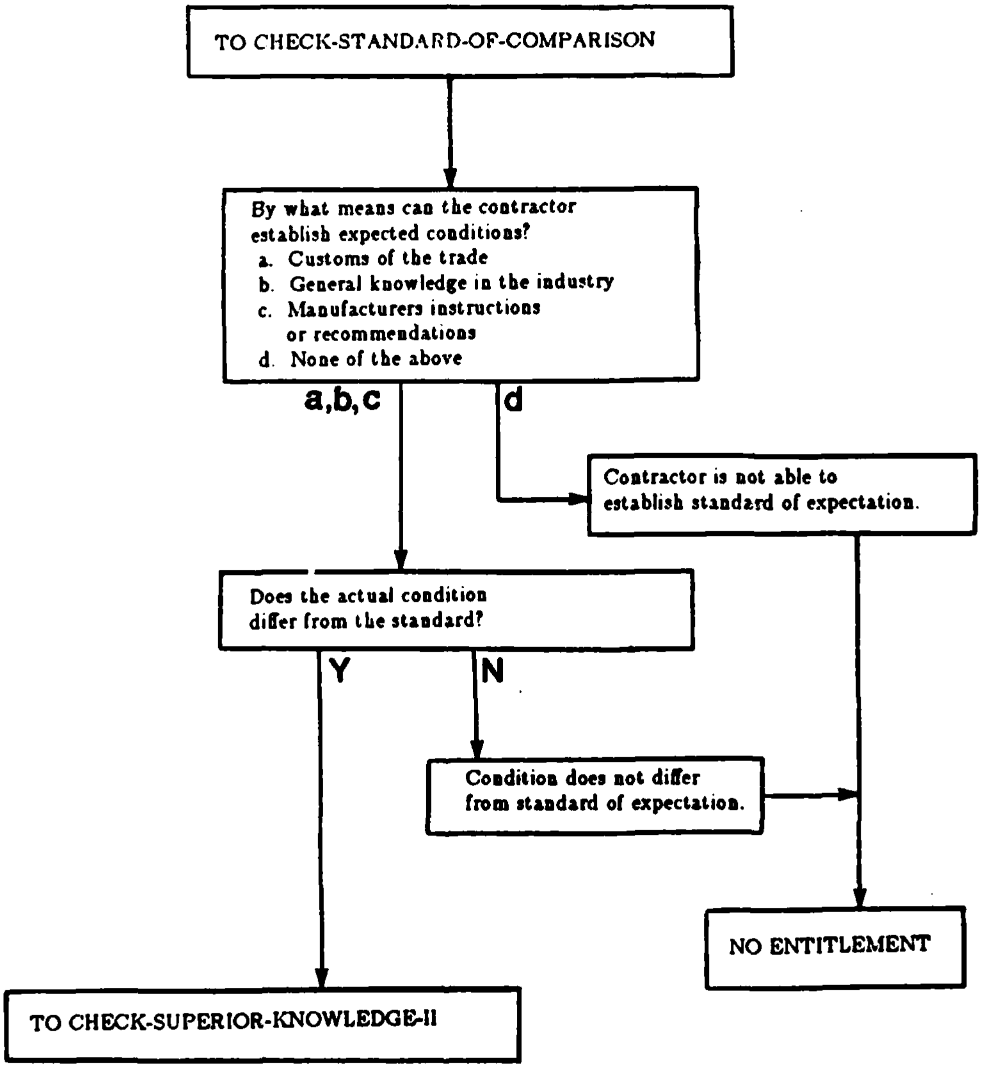


Figure A.14 Standard Conditions

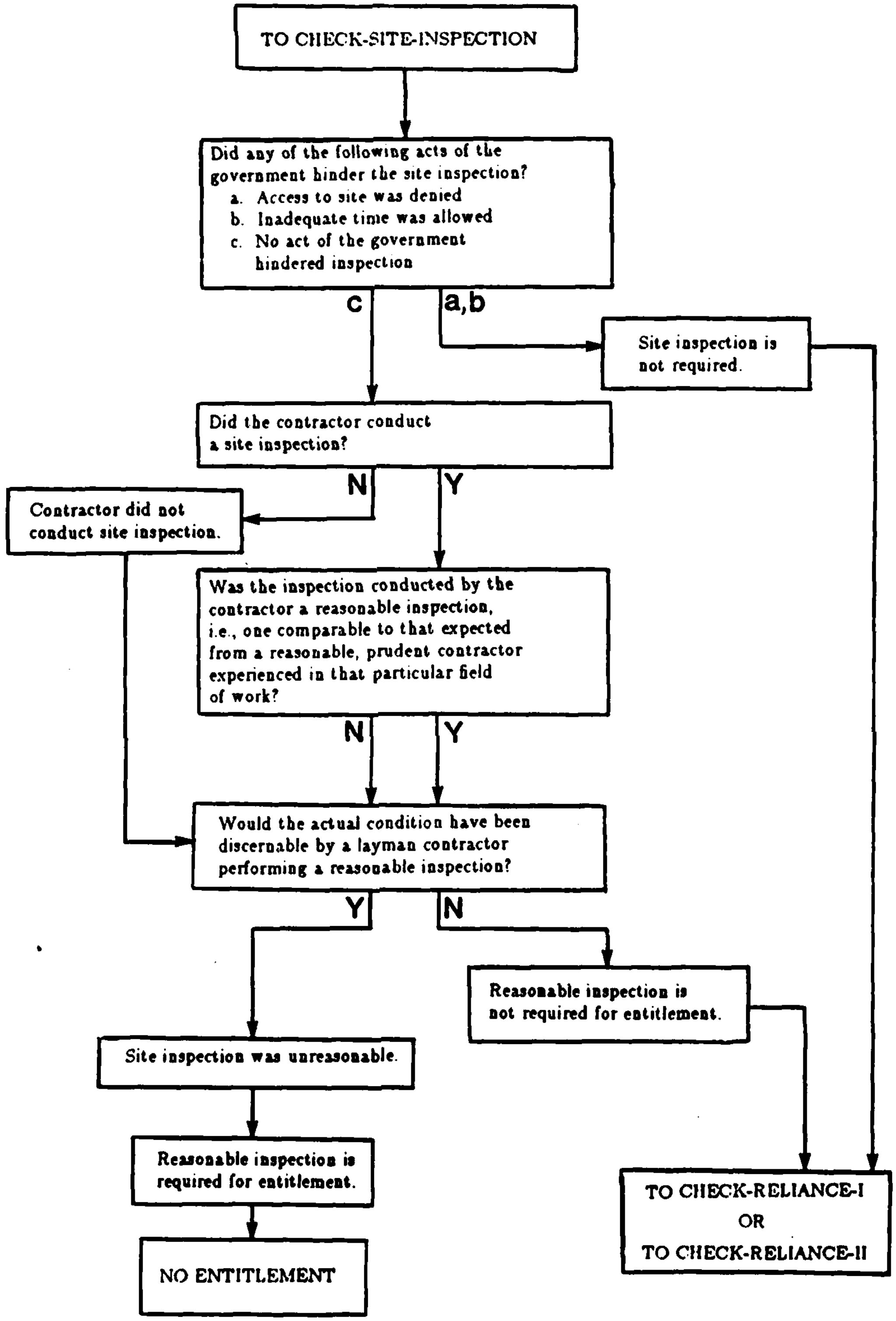


Figure A.15 Site Inspection

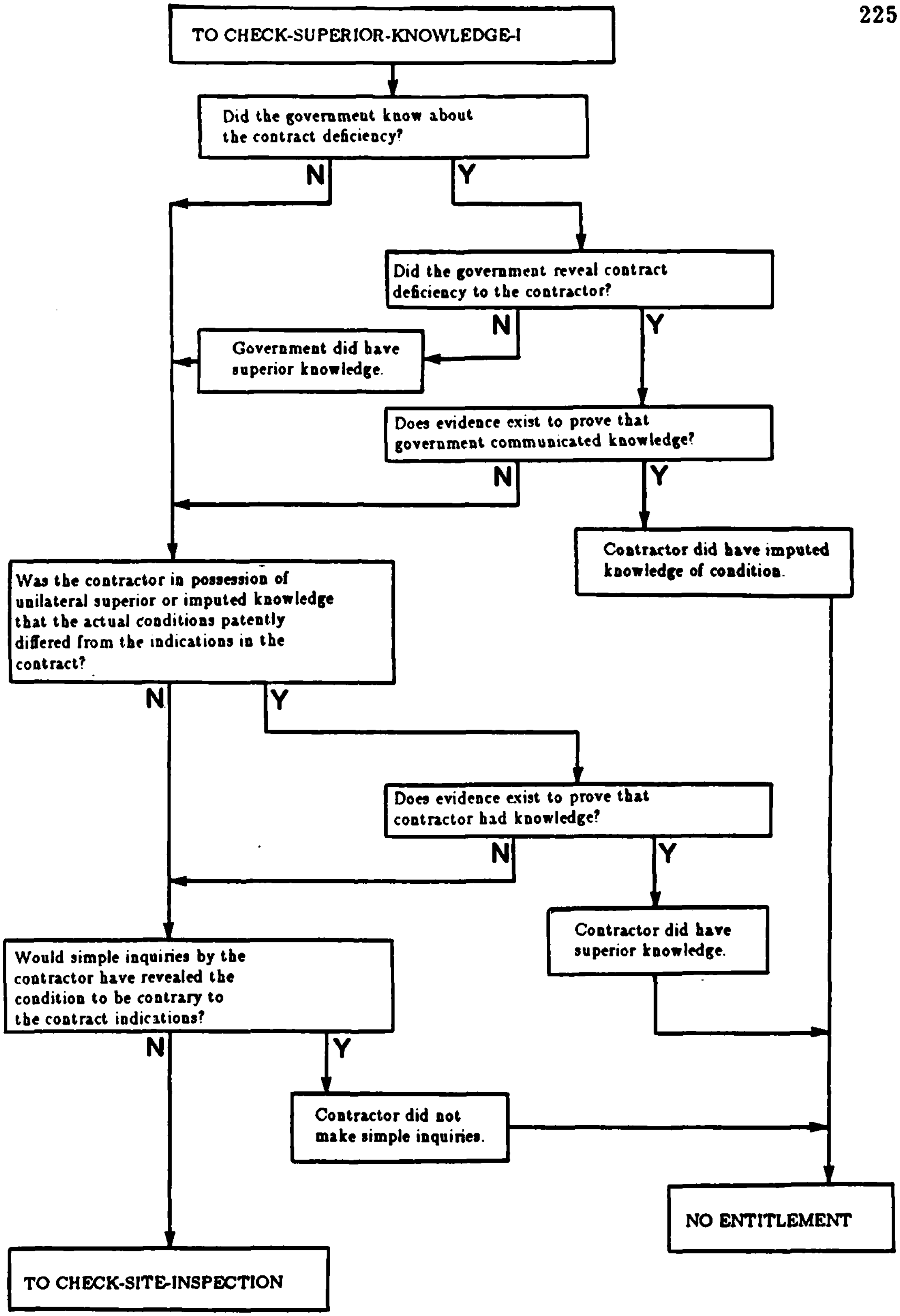


Figure A.16 Superior Knowledge-I

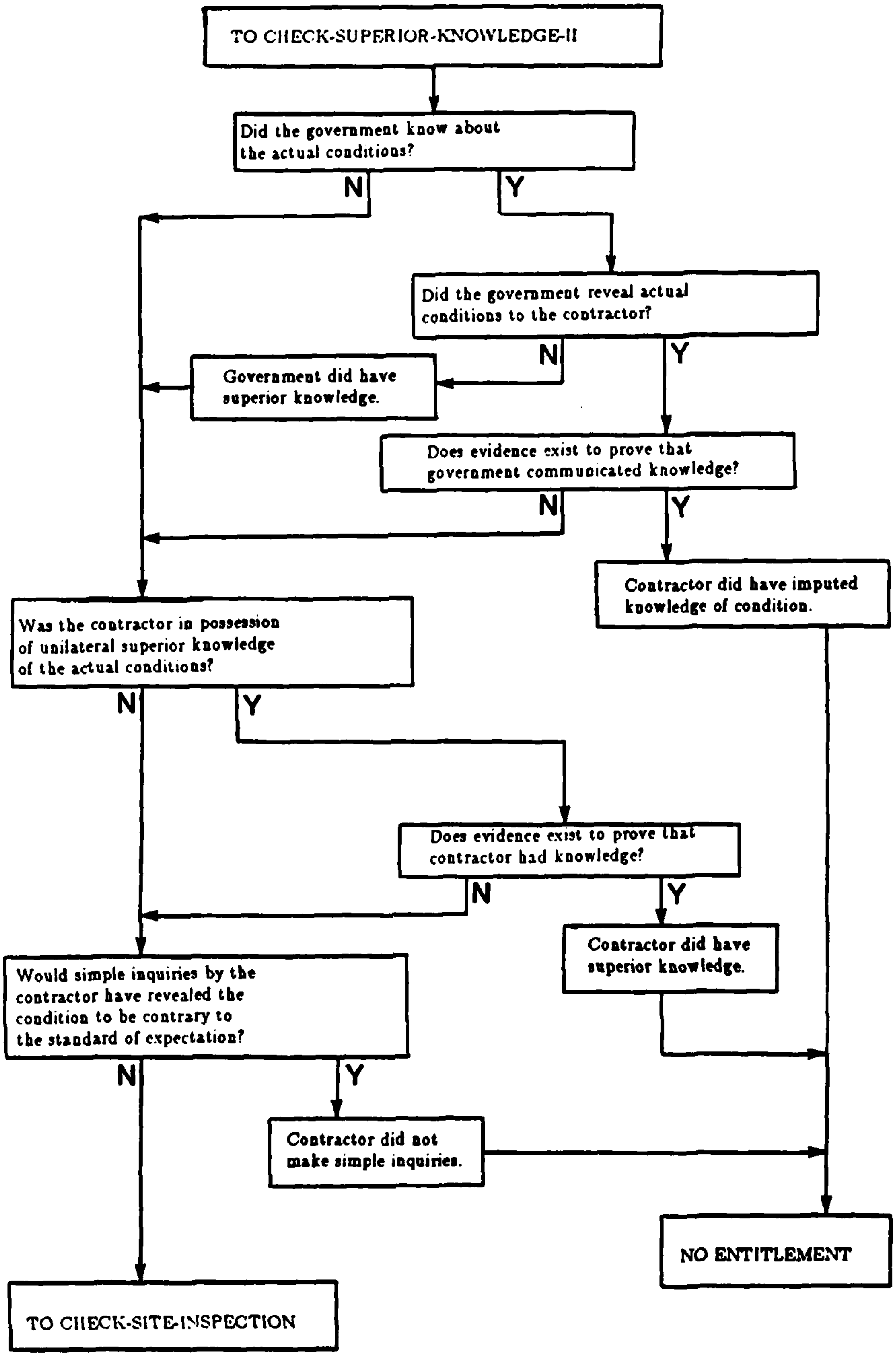


Figure A.17 Superior Knowledge-II

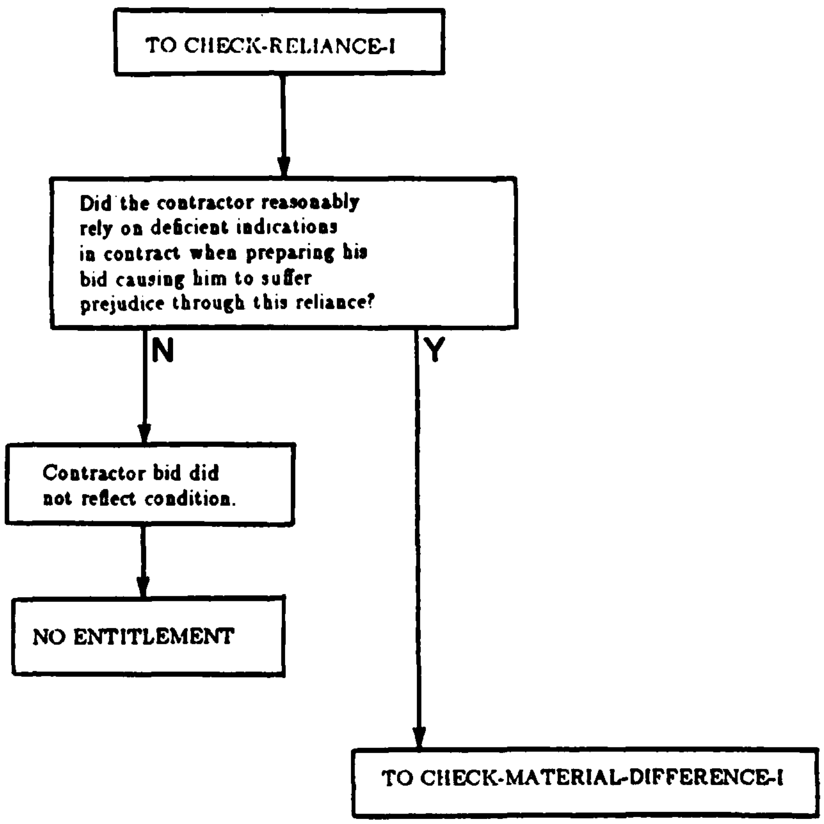


Figure A.18 Reliance I

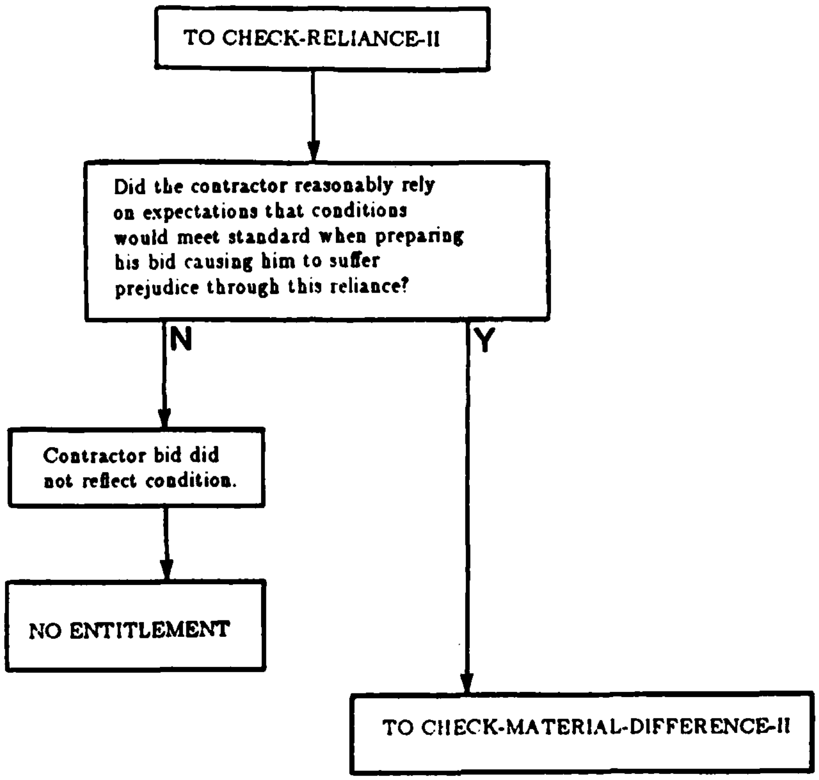


Figure A.19 Reliance II

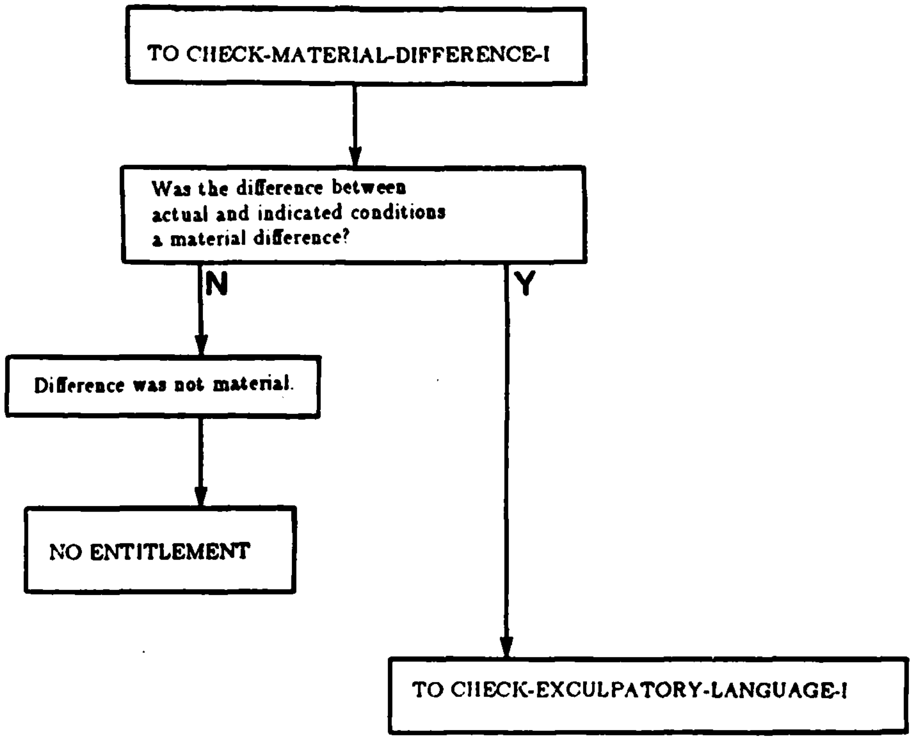


Figure A.2? Material Difference-I

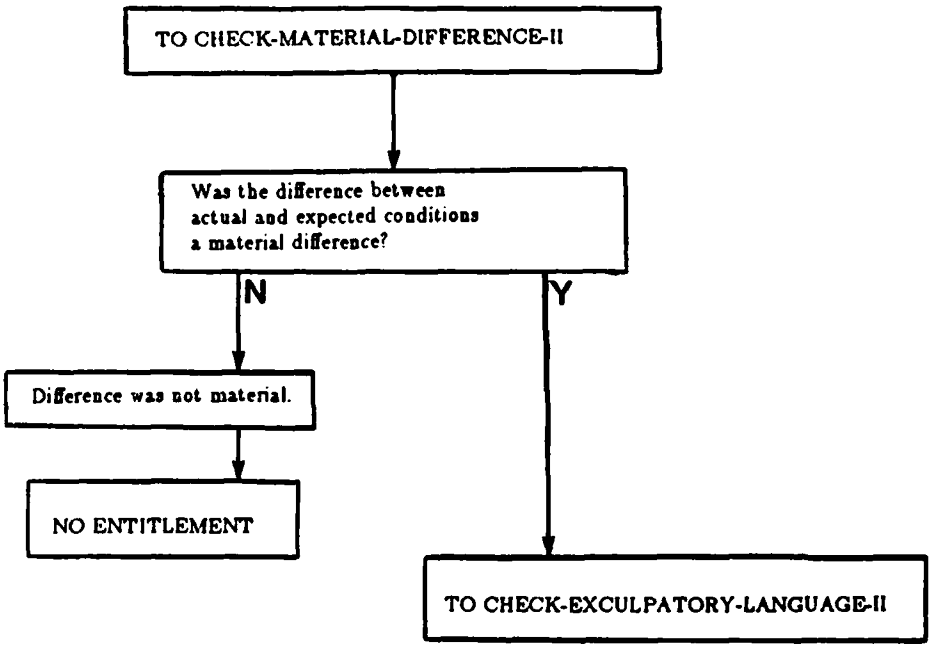


Figure A.21 Material Difference-II

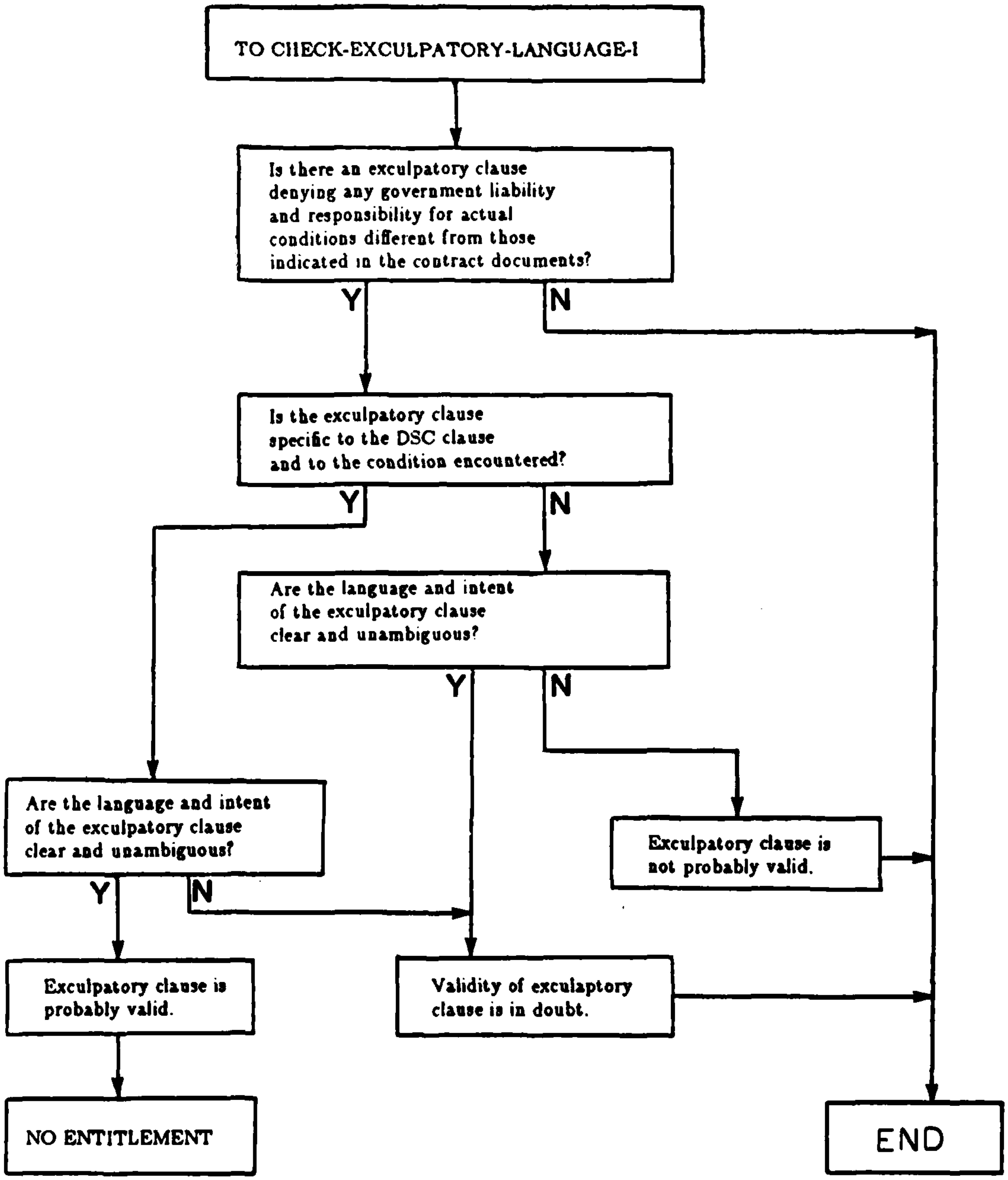


Figure A.22 Exculpatory Language-I

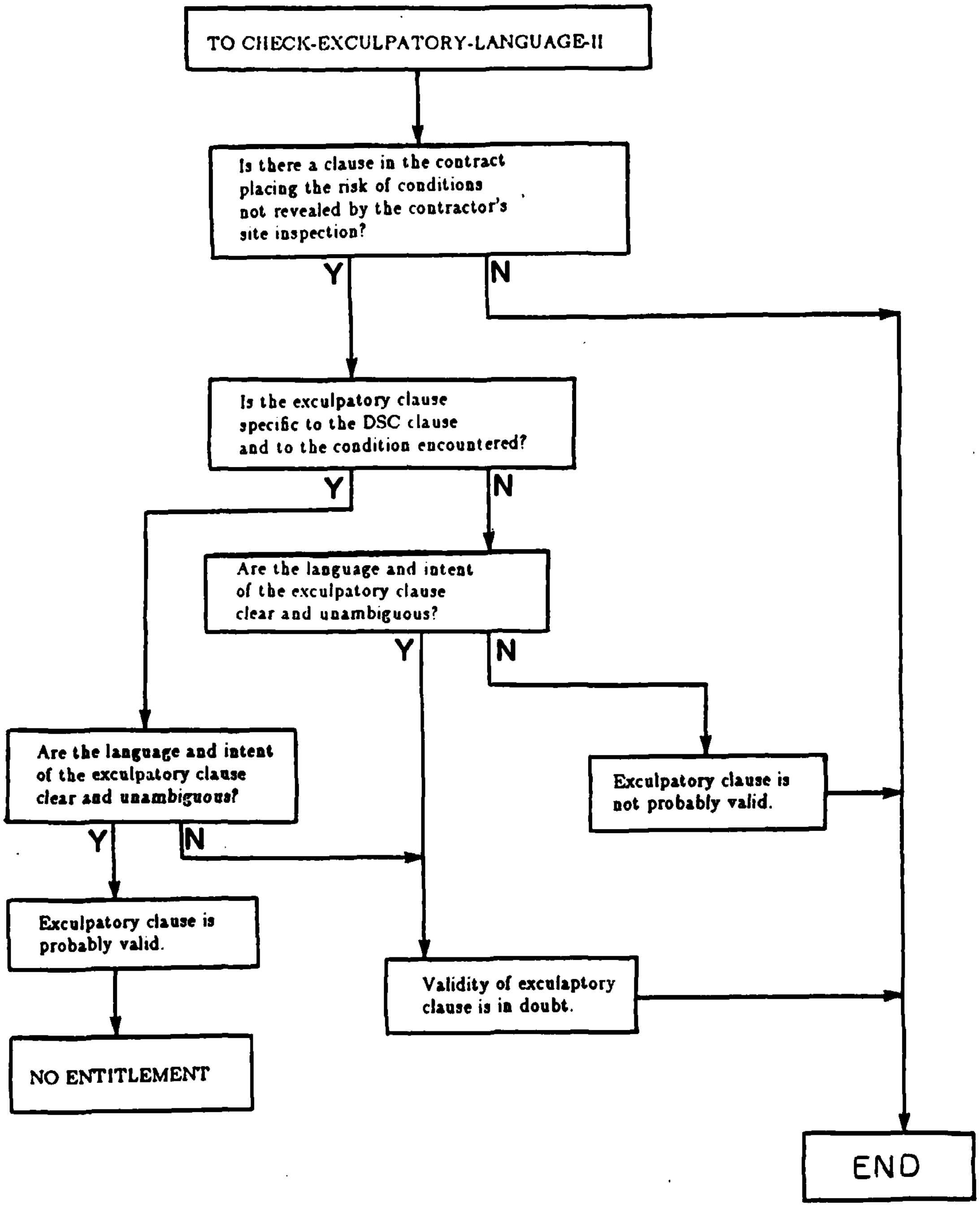


Figure A.23 Exculpatory Language-II

APPENDIX B

THE DSCAS PROGRAM

[: CONCLUDE1 parsed Thu Dec 1 14:11:14 1983 by Kruppen :]

```
To draw-DSC-assertion-conclusions:

[1] Do nothing.

End.


To draw-final-payment-conclusions:

[1] If ('DSC was asserted'
              is true in answers
        or 'DSC was asserted'
                   is true in assumptions)
        and 'final-payment was made'
                   is true in answers,
        go add 'contractor did probably concede rights-to-claim'
                                  to reports.

End.


To draw-form-of-notice-conclusions:

[1] Do nothing.

End.


To draw-promptness-conclusions:

[1] Do nothing.

End.


To draw-responsible-receiver-conclusions:

[1] Do nothing.

End.


To draw-prejudice-conclusions:

[1] Do nothing.

End.


To draw-obligation-conclusions:
```

[1] If 'government did expressly assume obligation-risk'
is true in answers
or 'contractor did expressly assume obligation-risk'
is true in answers,
go add 'contract did assign obligation-risk' to reports.

End.

To draw-excluded-conditions-conclusions:

[1] If 'condition is not directly-related-to-physical-conditions'
is true in answers
or 'condition is not static physical part of-work-site'
is true in answers,
go add 'condition is not considered to-be DSC' to reports.

End.

```
[: CONCLUDE2 parsed Thu Dec  1 14:15:45 1983 by Kruppen :]

To draw-after-bid-conditions-conclusions:

[1] If ('condition did not result from
            interacting-non-compensable-and-physical-factors'
                          is true in answers
      or 'condition did not result from
            interacting-non-compensable-and-physical-factors'
                          is true in assumptions)
      and ('condition did result from act-of-god'
                       is true in answers
        or 'condition did result from act-of-third-party'
                       is true in answers),
      go add 'condition is excluded condition' to reports.

[2] If 'condition did result from act-of-contractor'
                 is true in answers
     or 'condition did result from fault-of-contractor'
                 is true in answers,
      activate answers
      and if 'condition did result from act-of-government'
                       is not provably true
       or 'condition did result from fault-of-government'
                       is not provably true,
            go add 'condition is excluded condition' to reports,
      and deactivate.

[3] If 'condition is differing quantity'
                 is true in answers
            and 'contract does have var-in-est-quant-clause'
                       is true in answers,
            go add 'condition is best claimed through
                       var-in-est-quaht-clause' to reports.

[4] If ('condition did not result from
                 interacting-non-compensable-and-physical-factors'
                              is true in answers
       or 'condition did not result from
                 interacting-non-compensable-and-physical-factors'
                                     is true in assumptions)
      and ('condition is not differing quantity'
                       is true in answers
          or 'condition is not differing quantity'
                                     is true in assumptions),
            go add 'condition is excluded condition' to reports.

End.
```

[: CONCLUDE3 parsed Thu Dec 1 14:22:21 1983 by Kruppen :]

```
To draw-express-implied-deficiencies-conclusions:

[1] Do nothing.

End.


To draw-latent-deficiencies-conclusions:

[1] If 'contract indications did not contain latent deficiencies'
                                          is true in answers
        or ('contract information did contain physical data'
                                          is true in answers
            and 'contract information did reveal deficiencies/conditions'
                                          is true in answers),
           go add 'contractor did fail to-heed contract indications'
                                          to reports.

End.


To draw-contract-indications-conclusions:

[1] If 'contract information did contain physical data'
                                          is true in answers
      and 'contract information did reveal deficiencies/conditions'
                                          is true in answers,
           go add 'contractor did fail to-heed contract indications'
                                          to reports.

End.


To draw-standard-of-comparison-conclusions:

[1] If 'expected conditions are not establishable'
                                   is true in answers,
      go add 'contractor is not able-to establish standard-of-expectation'
                                   to reports.

[2] If ('expected conditions are establishable from customs-of-trade'
                                   is true in answers
       or 'expected conditions are establishable from
                            general-knowledge-in-industry'
                                   is true in answers
        or 'expected conditions are establishable from
                            manufacturers-recommendations'
                                   is true in answers)
      and 'condition does not differ from standard-of-expectation'
                                   is true in answers,
```

go add 'condition does not differ from standard-of-expectation' to reports.

End.

[: CONCLUDE4 parsed Thu Dec 1 14:28:03 1983 by Kruppen :]

To draw-superior-knowledge-I-conclusions:

[1] If 'government did reveal contract-deficiency'
is true in answers
and 'proof does indicate government-communicated-knowledge'
is true in answers,
go add 'contractor did have imputed knowledge-of-condition'
to reports.

[2] If 'contractor was aware-of contract-deficiency'
is true in answers
and 'proof does indicate contractor-had-knowledge'
is true in answers,
go add 'contractor did have superior knowledge' to reports.

[3] If 'simple inquiry did give potential to-know contrary conditions'
is true in answers,
go add 'contractor did not make simple inquiries'
to reports.

End.

To draw-superior-knowledge-II-conclusions:

[1] If 'government did reveal actual-conditions'
is true in answers
and 'proof does indicate government-communicated-knowledge'
is true in answers,
go add 'contractor did have imputed knowledge-of-condition'
to reports.

[2] If 'contractor was aware-of actual-conditions'
is true in answers
and 'proof does indicate contractor-had-knowledge'
is true in answers,
go add 'contractor did have superior knowledge' to reports.

[3] If 'simple inquiry did give potential to-know contrary conditions'
is true in answers,
go add 'contractor did not make simple inquiries'
to reports.

End.

To draw-site-inspection-conclusions:

[1] If 'reasonable inspection did have potential-to-reveal condition'
is true in answers,

go add 'reasonable inspection is required-for-entitlement'
to reports.

End.

[: CONCLUDE5 parsed Sat Nov 19 13:04:10 1983 by Kruppen :]

```
To draw-reliance-I-conclusions:

[1] If 'contractor did not suffer prejudice through reliance'
                        is true in answers,
     go add 'contractor bid did not reflect condition'
                           to reports.

End.

To draw-reliance-II-conclusions:

[1] If 'contractor did not suffer prejudice through reliance'
                        is true in answers,
     go add 'contractor bid did not reflect condition'
                           to reports.

End.

To draw-material-difference-I-conclusions:

[1] If 'difference was not material'
                   is true in answers,
     go add 'difference was not material' to reports.

End.

To draw-material-difference-II-conclusions:

[1] If 'difference was not material'
                   is true in answers,
     go add 'difference was not material' to reports.

End.

To draw-exculpatory-language-I-conclusions:

[1] If ('exculpatory-clause is specific-to DSC clause'
                   is true in answers
      or 'exculpatory-clause is specific-to DSC clause'
                   is true in assumptions)
      and ('exculpatory-clause is clear-and-unambiguous'
                   is true in answers
       or 'exculpatory-clause is clear-and-unambiguous'
                   is true in assumptions),
     go add 'exculpatory-clause is probably valid' to reports.
```

End.

To draw-exculpatory-language-II-conclusions:

```
[1] If ('exculpatory-clause is specific-to DSC clause'
                    is true in answers
        or 'exculpatory-clause is specific-to DSC clause'
                    is true in assumptions)
        and ('exculpatory-clause is clear-and-unambiguous'
                    is true in answers
          or 'exculpatory-clause is clear-and-unambiguous'
                    is true in assumptions),
       go add 'exculpatory-clause is probably valid' to reports.
```

End.

```
[: DRIVERS parsed Tue Dec  6 02:20:06 1983 by Kruppen :]

[******************************** DRIVERS ***********************************]

[*************************** Mainline Driver ******************************]

To check-complete-analysis:

Execute cyclically.

[1] Assert complete-analysis was run.

[2] Unless final-payment was checked,
        if entitlement is still-probable,
           go check-DSC-assertion
           and go check-final-payment
           and go draw-final-payment-conclusions.

[3] Unless responsible-receiver was checked,
        if entitlement is still-probable,
              go check-form-of-notice
              and go check-promptness
              and go check-responsible-receiver.

[4] If responsible-receiver was checked,
          activate reports
          and if (there is no in_doubt
                  or notice requirement was waived),
                    assert contractor did comply-fully
                             with notice-requirement
                    and deactivate,
                 otherwise,
                deactivate.

[5] Unless prejudice was checked,
       if 'contractor did comply-fully with notice-requirement'
                                   is true in reports,
           do nothing,
        otherwise if entitlement is still-probable,
          go check-prejudice.

[6] Unless obligation was checked,
        if entitlement is still-probable,
              go check-obligation
              and go draw-obligation-conclusions.

[7] Unless excluded-conditions was checked,
        if entitlement is still-probable,
           go check-excluded-conditions
           and go draw-excluded-conditions-conclusions.

[8] Unless after-bid-conditions was checked,
```

```
    if entitlement is still-probable
        and 'condition did occur after contract award'
                           is true in answers,
        go check-after-bid-conditions
        and go draw-after-bid-conditions-conclusions.

[9] Unless express-implied-deficiencies was checked,
    if entitlement is still-probable,
        go check-express-implied-deficiencies.

[10] If 'latent-deficiencies was checked' is not provably true
    and 'contract-indications was checked' is not provably true
    and entitlement is still-probable,
        if 'contract did contain indications concerning condition'
                           is true in reports,
            go check-latent-deficiencies
            and go draw-latent-deficiencies-conclusions,
        otherwise if 'contract did not contain
                           indications concerning condition'
                           is true in reports,
            go check-contract-indications
            and go draw-contract-indications-conclusions.

[11] Unless standard-of-comparison was checked,
    if entitlement is still-probable
        and contract-indications was checked,
        go check-standard-of-comparison
        and go draw-standard-of-comparison-conclusions.

[12] Unless knowledge-I was checked,
    if entitlement is still-probable
        and latent-deficiencies was checked,
            go check-superior-knowledge-I
            and go draw-superior-knowledge-I-conclusions.

[13] Unless knowledge-II was checked,
    if entitlement is still-probable
        and standard-of-comparison was checked,
            go check-superior-knowledge-II
            and go draw-superior-knowledge-II-conclusions.

[14] Unless site-inspection was checked,
    if entitlement is still-probable,
            go check-site-inspection
            and go draw-site-inspection-conclusions.

[15] If 'reliance-I was checked' is not provably true
    and 'reliance-II was checked' is not provably true
    and entitlement is still-probable
    and ('site-inspection is not required' is true in reports
        or 'reasonable inspection did not reveal condition'
                                   is true in reports
        or 'reasonable inspection is not required-for-entitlement'
```

```
                                  is true in reports),
                if latent-deficiencies was checked,
                        go check-reliance-I
                        and go draw-reliance-I-conclusions,
                 otherwise,
                        go check-reliance-II
                        and go draw-reliance-II-conclusions.

[16] If 'material-difference-I was checked' is not provably true
        and 'material-difference-II was checked' is not provably true
        and entitlement is still-probable,
                if knowledge-I was checked,
                        go check-material-difference-I
                        and go draw-material-difference-I-conclusions,
                 otherwise if knowledge-II was checked,
                        go check-material-difference-II
                        and go draw-material-difference-II-conclusions.

[17] If 'exculpatory-language-I was checked' is not provably true
        and 'exculpatory-language-II was checked' is not provably true
        and entitlement is still-probable,
                if material-difference-I was checked,
                        go check-exculpatory-language-I
                        and go draw-exculpatory-language-I-conclusions,
                 otherwise if material-difference-II was checked,
                        go check-exculpatory-language-II
                        and go draw-exculpatory-language-II-conclusions.

[18] Go check-entitlement
        and if 'reason_for no entitlement does exist' is true in reports,
                        return.

[19] If analysis is complete,
                        return.

End.


[*********************** Drivers for individual modules **********************]

To check-DSC-assertion:

[1] If DSC-assertion was checked,
        return,
    otherwise,
        assert DSC-assertion was checked.

[2] If 'DSC was asserted' is true in answers,
            send {2 lines,"The differing site condition has been asserted.",cr}
            and activate answers
            and send {"The date of assertion of the differing site ",cr,
            "condition is ",the date-of-DSC-assertion,".",2 lines}
            and deactivate,
```

```
        otherwise if 'DSC was not asserted' is true in answers,
          send {1 line,"I'm sorry, but I can't help you with your problems...",
                                2 lines}
          and go wrap-up,
        otherwise if 'if-DSC was asserted' is true in unknowns,
          go unknown-DSC-assertion,
        otherwise,
          go determine-if-DSC-was-asserted.

End.


[********************************]

To check-final-payment:

[1] Assert final-payment was checked.

[2] If 'final-payment was made' is true in answers,
      send {3 lines,"The final payment was made.",1 line}
      and activate answers
      and send {1 line,"The date of final payment was ",
                    the date-of-final-payment,".",cr}
      and deactivate
      and deny entitlement is still-probable
      and assert entitlement is not still-probable,
    otherwise if 'final-payment was not made' is true in answers,
      send {3 line,"The final payment was not made.",1 line},
    otherwise if 'if-final-payment was made' is true in unknowns,
      go unknown-final-payment,
    otherwise,
      go determine-if-final-payment-was-made.

End.


[********************************]

To check-form-of-notice:

[1] If form-of-notice was checked,
        return,
    otherwise,
        assert form-of-notice was checked.

[2] Activate answers
    and if form-of-notice was written,
      go add 'contractor did comply with written-notice-requirement'
                                                to reports
      and send {2 lines,"The contractor did comply with the notice ",cr,
                    "requirements.",cr,"The written notice was given on ",
                    the date-of-written-notice,".",cr}
      and deactivate,
```

```
    otherwise if form-of-notice was oral,
      go add 'contractor did constructively comply with notice-requirements'
                                                              to reports
      and send {2 lines,"The contractor gave oral notice, therefore,the",cr,
                       "contractor constructively complied with the ",cr,
                  "notice requirement.",cr,
                       "The oral notice was given on ",
                            the date-of-oral-notice,".",cr}
      and deactivate,
    otherwise if form-of-notice was other,
     deactivate,
    otherwise if 'form-of-notice is unknown' is true in unknowns,
     deactivate
     and go unknown-form-of-notice,
    otherwise,
     deactivate
     and go determine-form-of-notice.

[3] If 'form-of-notice was other' is true in answers,
     if 'notice-requirement was not waived' is true in answers,
        go add 'notice-requirement was not waived' to reports
        and go add 'actual-occurrence-of-notice is an in-doubt'
                                                   to reports
        and send {1 line,"Actual occurrence of notice to government is in doubt.",cr},
      otherwise if 'emergency-situation was encountered' is true in answers,
           go add 'notice-requirement was waived' to reports
           and send {1 line,"An emergency-situation was encountered by the ",cr,
                    "contractor which required immediate action.",cr},
      otherwise if 'same-situation was encountered previously'
                                            is true in answers,
           go add 'notice-requirement was waived' to reports
           and send {1 line,"The government received notice of the same",cr,
                    "condition at a previous time on this job.",cr},
      otherwise if 'reason-for-notice-waiver does exist' is true in answers,
           activate answers
           and send {1 line,"Other events occurred which led to the belief",cr,
                    "that the notice requirement had been waived.",1 line,
                    "The other event was:  ",the reason-for-notice-waiver,
                                            2 lines}
           and deactivate
           and go add 'notice-requirement was waived' to reports,
      otherwise if 'if-notice requirement was waived' is true in unknowns,
           go unknown-other-notice,
      otherwise,
           go determine-other-notice.

End.


[***********************************************]

To check-promptness:
```

```
[1] If promptness was checked,
            return,
        otherwise,
            assert promptness was checked.


[2] If 'notice was given before contractor disturbed condition'
                                    is true in answers
    and 'notice was given before remedial work performed'
                                    is true in answers,
    send {1 line,"The notice was given before disturbing the condition",
                cr,"and before performing the work.",cr,
                "Therefore, notice was prompt.",cr},
    otherwise if 'notice was given before contractor disturbed condition'
                                    is true in answers,
    send {1 line,
        "The notice was given before disturbing the condition.",cr}
    and go add 'promptness_of_notice is an in_doubt' to reports,
    otherwise if 'notice was given before remedial work performed'
                                    is true in answers,
    send {1 line,"The notice was given before performing the work.",cr}
    and go add 'promptness_of_notice is an in_doubt' to reports,
    otherwise if 'notice was not given before remedial work performed'
                                    is true in answers,
    send {1 line,"The notice was given after disturbing the condition",cr,
        "or after performing remedial work.",cr}
    and go add 'promptness_of_notice is an in_doubt' to reports,
    otherwise if 'if-notice was given before condition disturbed'
                                    is true in unknowns,
    go unknown-promptness,
    otherwise,
    go determine-promptness.

[3] If 'contractor did not have potential-to-know condition sooner'
                                    is true in answers,
    send {1 line,"The contractor should not have been aware",cr,
                "of the condition sooner.",cr}
    and go add 'notice was prompt' to reports,
  otherwise if 'contractor did have potential-to-know condition sooner'
                                    is true in answers,
    send {1 line,"The contractor should have been aware of the ",cr,
                "condition sooner.",cr}
    and go add 'promptness_of_notice is an in_doubt' to reports,
  otherwise if 'if-contractor did have potential-to-know condition sooner'
                                            is true in unknowns,
    go unknown-potential-promptness,
  otherwise,
    go determine-potential-promptness.

End.
```

```
[****************************************************]

To check-responsible-receiver:

[1] If responsible-receiver was checked,
      return,
    otherwise,
      assert responsible-receiver was checked.

[2] If 'notice was received-by contracting officer' is true in answers,
      send {3 lines,"Contracting officer received notice of DSC.",cr}
      and go add 'notice was given to responsible receiver' to reports
      and go add 'government did receive notice-of-condition' to reports
      and return,
    otherwise if 'notice was received-by authorized representative '
                                                  is true in answers,
      send {3 lines,"Notice of DSC was given to an authorized ",cr,
                    "representative of the CO.",cr}
      and go add 'notice was given to responsible receiver' to reports
      and go add 'government did receive notice-of-condition' to reports
      and return,
    otherwise if 'notice was received-by other government employee'
                                    is true in answers,
      send {3 lines,"Notice of the DSC was given to some other ",cr,
                    "government employee.",cr},
    otherwise if 'notice was not received-by government' is true in answers,
      send {3 lines,"Government did not receive notice of condition.",cr}
      and go add 'receipt-of-notice is an in_doubt' to reports
      and return,
    otherwise if 'to-whom was notice given' is true in unknowns,
      go unknown-receiver,
    otherwise,
      go determine-responsible-receiver.

[3] If 'notice was received-by other government employee' is true in answers,
      if 'government employee did understand implications' is true in answers,
        go add 'notice was given-to responsible receiver' to reports
        and send {2 lines,"The other government employee was cognisant",cr,
                  "of the implications of the condition.",cr},
      otherwise if 'government employee did not understand implications'
                                            is true in answers,
        send {2 lines,"The other government employee was not cognisant",cr,
                  "of the implications of the condition, therefore,",cr,
                  "government receipt of notice is in doubt.",cr}
        and go add 'receipt-of-notice is an in_doubt' to reports
        and return,
      otherwise if 'if-government employee did understand implications'
                                            is true in unknowns,
        go unknown-position,
      otherwise,
        go determine-position.

[4] If 'notice was received-by other government employee' is true in answers
```

```
    and ('government employee did understand implications'
                                   is true in answers
       or 'government employee did understand implications'
                                   is true in assumptions),
    if 'government employee did communicate DSC awareness to-CO'
                                   is true in answers,
         go add 'government was aware of DSC' to reports
         and send {1 line,
         "The government employee to whom notice of the DSC was ",cr,
           "given did communicate his awareness of the condition to",cr,
           "the CO.",cr}
         and go add 'government did receive notice-of-condition' to reports,
     otherwise if 'government employee did not communicate DSC awareness to-CO'
                                   is true in answers,
         do nothing,
     otherwise if 'if-government employee did communicate DSC awareness to-CO'
                                   is true in unknowns,
         go unknown-communication-of-notice,
     otherwise,
         go determine-communication-of-notice.

[5] If 'government employee did not communicate DSC awareness to-CO'
                                   is true in answers,
    if 'proof-of DSC notice does exist' is true in answers,
     send {1 line,"The government employee to whom notice of the DSC was ",
           cr,"given did not communicate his awareness of the",cr,
         "condition to the CO. However, proof that the notice",cr,
         "was given does exist.", cr}
       and go add 'government was aware of DSC' to reports,
     otherwise if 'proof-of DSC notice does not exist' is true in answers,
         send {1 line,"The government employee to whom notice of the DSC",cr,
         "was given did not communicate his awareness of the ",cr,
         "condition to the CO and there is no proof that notice",cr,
         "was given, therefore, government awareness of the DSC",cr,
         "is in doubt.",cr}
       and go add 'government awareness_of_DSC is an in_doubt' to reports,
     otherwise if 'if-proof-of DSC notice does exist' is true in unknowns,
         go unknown-proof-of-DSC-notice,
     otherwise,
     go determine-proof.

End.


{*******************************************}

To check-prejudice:

[1] Assert prejudice was checked.

[2] If 'passage-of-time did obscure evidence' is true in answers,
         send {3 line,"Evidence against the claim was obscurred due to",cr,
              "the passage of time.",cr}
```

```
                  and go add 'passage-of-time did cause prejudice' to reports,
          otherwise if 'passage-of-time did not obscure evidence' is true in answers,
                  send {3 line,"Evidence against the claim was not obscurred due",cr,
                         "to the passage of time.",cr},
          otherwise if 'if-passage-of-time did obscure evidence' is true in unknowns,
                  go unknown-prejudice-from-passage-of-time,
          otherwise,
                  go determine-prejudice-from-passage-of-time.

[3] If 'improper-notice did cause additional cost' is true in answers,
                  send {1 line,"Improper notice resulted in a more expensive",cr,
                  "resolution of DSC.",cr}
                  and go add 'additional-cost did cause prejudice' to reports,
         otherwise if 'improper-notice did not cause additional cost'
                                          is true in answers,
                  send {1 line,"Improper notice did not result in a more expensive ",cr,
                       "resolution of DSC.",cr},
         otherwise if 'if-improper-notice did cause additional cost'
                                          is true in unknowns,
                  go unknown-prejudice-from-late-notice,
         otherwise,
                  go determine-prejudice-from-late-notice.

[4] If 'something-else did cause prejudice' is true in answers,
                  send {1 line,"Some event other than obscurred evidence or ",cr,
                        "increased cost has caused the government to be",cr,
                             "prejudiced.",cr}
                  and go add 'something-else did cause prejudice' to reports,
         otherwise if 'nothing-else did cause prejudice' is true in answers,
                  send {1 line,
                  "The government was not prejudiced in any additional way.",cr},
         otherwise if 'if-anything-else did cause prejudice' is true in unknowns,
                  go unknown-other-prejudice,
         otherwise,
                  go determine-other-prejudice.

[5] If ('passage-of-time did obscure evidence' is true in answers
       or 'improper-notice did cause additional cost' is true in answers
        or 'something-else did cause prejudice' is true in answers),
          send {2 lines,"The government did suffer prejudice.",cr}
          and go add 'government did suffer prejudice' to reports,
     otherwise if ('passage-of-time did not obscure evidence'
                                     is true in assumptions
          or 'improper-notice did not cause additional cost'
                                     is true in assumptions
        or 'nothing-else did cause prejudice' is true in assumptions),
       go add 'government prejudice is uncertain' to reports
       and send {2 lines,"It is uncertain whether or not the government can",cr,
       "prove that it suffered prejudice from inadequacy of",cr,
       "or lack of notice.",cr}
       and return,
     otherwise if 'passage-of-time did not obscure evidence'
                                     is true in answers
```

```
and 'improper-notice did not cause additional cost'
                                is true in answers
and 'nothing-else did cause prejudice' is true in answers,
go add 'government did not suffer prejudice' to reports
and send {2 lines,"Government can't prove that it suffered prejudice.",cr}
and return.

[6] If 'government did suffer prejudice' is true in reports,
    if 'defense-against claim was made impossible' is true in answers,
      send {1 line,"Government defense against the claim was made",cr,
        "impossible by inadequacy of or lack of notice, this",cr,
        "plus the resulting prejudice to the government did ",cr,
        "render entitlement to DSC claim unlikely.",cr,
        "However, I will continue our analysis...",cr}
      and return,
    otherwise if 'defense-against claim was not made impossible'
                                is true in answers,
        send {1 line,"Improper notice slightly increased the difficulty of",cr,
                "defending against the claim.",cr},
    otherwise if 'if-difficulty-of-defending-against claim was increased'
                                is true in unknowns,
        go unknown-difficulty-of-defending-claim,
    otherwise,
        go determine-difficulty-of-defending-claim.

[7] If 'government did suffer prejudice' is true in reports
    and ('defense-against claim was not made impossible'
                                is true in answers
        or 'defense-against claim was not made impossible'
                                is true in assumptions),
    if 'additional-proof does exist to-prove-entitlement'
                                is true in answers,
          send {1 line,"There is sufficient proof to prove claim ",cr,
            "to entitlement.",2 lines,
            "The contractor may be entitled to claim in spite",cr,
            "of the inadequacy of or lack of notice and in ",cr,
            "spite of the resulting prejudice to the government.",
                cr},
    otherwise if 'additional-proof does not exist to-prove-entitlement'
                                is true in answers,
          send {1 line,"There is not sufficient proof to prove claim ",cr,
                "to entitlement.",2 lines,
            "Since there was inadequate notice or lack of ",cr,
            "notice plus the resulting prejudice to the ",cr,
            "government entitlement is unlikely. ",cr,
            "However, I will continue our analysis...",cr},
    otherwise if 'if-additional-proof does exist to-prove-entitlement'
                                is true in unknowns,
            go unknown-proof-of-claim,
    otherwise,
            go decide-proof-of-claim.

End.
```

[: DRIVERS2 parsed Thu Dec 1 21:23:18 1983 by Kruppen :]

```
[***************************************]

To check-obligation:

[1] Assert obligation was checked.

[2] If 'government did expressly assume obligation-risk' is true in answers,
      send {3 lines,"The government expressly assumed the obligation or",cr,
            "risk of this condition, therefore, relief may come from",cr,
            "the clause assigning this obligation or risk.",cr}
      and deny entitlement is still-probable
      and assert entitlement is not still-probable
      and return,
    otherwise if 'contractor did expressly assume obligation-risk'
                                          is true in answers,
      send {3 lines,"The contractor has assumed the obligation or risk of",cr,
            "the condition.",cr}
      and deny entitlement is still-probable
      and assert entitlement is not still-probable
      and return,
    otherwise if 'confusion-about-who did assume risk' is true in answers,
      send {3 lines,"There seems to be confusion as to who (the government",cr,
            "or the contractor) expressly assumed the risk of",cr,
            "the condition.",cr}
      and go unknown-obligation,
    otherwise if 'government did not expressly assume obligation-risk'
                                          is true in answers
      and 'contractor did not expressly assume obligation-risk'
                                          is true in answers,
      send {3 lines,"Neither the government nor the contractor have assumed ",cr,
                "obligation or risk for the condition.",cr}
      and go add 'contract did not assign obligation-risk' to reports,
    otherwise if 'which-party did assume risk' is true in unknowns,
      go unknown-obligation,
    otherwise,
      go determine-obligation.

End.

[***************************************]

To check-excluded-conditions:

[1] Assert excluded-conditions was checked.

[2] If 'condition is directly-related-to-physical-conditions'
                                          is true in answers,
       send {3 lines,"The condition is directly related to the physical ",cr,
                "conditions at the site.",cr},
    otherwise if 'condition is not directly-related-to-physical-conditions'
```

```
                                    is true in answers,
   send {3 lines,"The condition is not directly related to the physical",cr,
         "conditions at the site.  The differing site conditions",cr,
         "claim is limited to conditions and causes that are ",cr,
         "directly related to the physical conditions at the site.",cr}
   and deny entitlement is still-probable
   and assert entitlement is not still-probable
   and return,
  otherwise if 'if-condition is directly-related-to-physical-conditions'
                                    is true in unknowns,
   go unknown-physical-conditions,
  otherwise,
   go determine-physical-conditions.

[3] If ('condition is directly-related-to-physical-conditions'
                                    is true in answers
     or 'condition is directly-related-to-physical-conditions'
                                    is true in assumptions),
   if 'condition is static physical part of-work-site' is true in answers,
        send {1 lines,"The condition is a static physical part of",cr,
              "the work site.",cr},
   otherwise if 'condition is not static physical part of-work-site'
                                    is true in answers,
        send {3 lines,"The condition is not a static physical part",cr,
                 "of the work site.  The differing site condition",cr,
                 "is limited to conditions which are a static",cr,
                 "physical of the work site.  Recovery may be",cr,
                 "possible through another clause or through breach.",cr}
        and deny entitlement is still-probable
        and assert entitlement is not still-probable
        and return,
   otherwise if 'if-condition is static physical part of-work-site'
                                    is true in unknowns,
        go unknown-static-physical-condition,
   otherwise,
        go determine-static-physical-condition.

[4] If ('condition is static physical part of-work-site' is true in answers
     or 'condition is static physical part of-work-site'
                                    is true in assumptions),
   if 'condition did occur before contract award' is true in answers,
        send {1 lines,"The condition did occur before the contract was ",cr,
        "awarded.",cr},
    otherwise if 'condition did occur after contract award'
                                     is true in answers,
        send {1 lines,"The condition occurred after the contract was awarded.",
               cr,"The differing site condition is usually limited to ",cr,
                 "those conditions occurring before the contract award.",
              1 line,"However, an exception may be allowed.",cr},
    otherwise if 'if-condition did occur before contract award'
                                     is true in unknowns,
        go unknown-condition-occurrence,
    otherwise,
```

```
        go determine-condition-occurrence.

End.

[****************************************]

To check-after-bid-conditions:

[1] Assert after-bid-conditions was checked.

[2] If ('condition did result from act-of-god' is true in answers
         or 'condition did result from act-of-government' is true in answers
        or 'condition did result from act-of-third-party' is true in answers
        or 'condition did result from act-of-contractor' is true in answers
         or 'condition did result from fault-of-contractor'
                                         is true in answers
          or 'condition did result from fault-of-government'
                                         is true in answers
           or 'condition did result from no-one' is true in answers
        or 'what-condition did result-from' is true in unknowns),
            deny cause-of-condition is unknown.

[3] If cause-of-condition is unknown,
            go determine-cause,
        otherwise if 'what-condition did result-from' is true in unknowns,
            go unknown-cause,
        otherwise,
            send {3 lines,"Thus far as to the cause of the condition we know that:",cr}
            and if 'condition did result from act-of-god' is true in answers,
                send {1 lines,"It was due to an act of God.",cr},
            and if 'condition did result from act-of-third-party'
                                          is true in answers,
                send {1 lines,"It was due to an act of a third party.",cr},
            and if 'condition did result from act-of-government'
                                          is true in answers,
                send {1 lines,"It was due to an act of the government.",cr},
            and if 'condition did result from fault-of-government'
                                          is true in answers,
                send {1 lines,"It was due to a fault of the government.",cr},
            and if 'condition did result from act-of-contractor'
                                          is true in answers,
                send {1 lines,"It was due to an act of the contractor.",cr},
            and if 'condition did result from fault-of-contractor'
                                          is true in answers,
                send {1 lines,"It was due to a fault of the contractor.",cr},
            and if 'condition did result from no-one' is true in answers,
                send {1 lines,"No party involved caused condition.",cr}.

[4] If 'condition did result from no-one' is true in answers
       and ('condition did result from act-of-god' is true in answers
           or 'condition did result from act-of-third-party'
                                         is true in answers
```

```
                or 'condition did result from act-of-government'
                                    is true in answers
                or 'condition did result from act-of-contractor'
                                    is true in answers
                or 'condition did result from fault-of-government'
                                    is true in answers
                or 'condition did result from fault-of-contractor'
                                    is true in answers),
            send {2 lines,"You have stated that no-one caused the condition and",cr,
                    "also stated that the condition was caused by one of",cr,
                    "the other choices.  To alleviate the conflict I will",cr,
                    "forget about your input indicating that no-one caused",cr,
                    "the condition.",cr}
            and go remove 'condition did result from no-one' from answers.

[5] If ('condition did result from act-of-contractor' is true in answers
        or 'condition did result from fault-of-contractor'
                                    is true in answers),
      if ('condition did result from act-of-government' is true in answers
        or 'condition did result from fault-of-government'
                                    is true in answers),
      go add 'condition was caused-by mutual acts' to reports
      and send {2 lines,"The condition resulted from mutual acts of",cr,
                "the government and the contractor.",cr}
      and return,
        otherwise,
      go add 'contractor did contribute to condition' to reports
      and deny entitlement is still-probable
      and assert entitlement is not still-probable
      and return.

[6] Activate answers
      and if ((condition did result from act-of-government
            or condition did result from fault-of-government)
              and condition did result from act-of-third-party)
              and 'condition did result from act-of-contractor'
                                    is not provably true
              and 'condition did result from fault-of-contractor'
                                    is not provably true
              and 'condition did result from act-of-god'
                                    is not provably true,
          (if government did have control over third party,
                send {1 lines,"The government did have control over",cr,
                        "the actions of the third party.",cr}
                and deactivate,
           otherwise if government did not have control over third party,
                send {1 lines,"The government did not have control over",
                        cr,"the actions of the third party.",cr}
                and deactivate,
           otherwise if 'if-government did have control over third party'
                                    is true in unknowns,
                deactivate
                and go unknown-government-control,
```

```
            otherwise,
                 deactivate
                 and go determine-government-control),
      otherwise,
           deactivate.

[7] If 'condition did result from act-of-third-party' is true in answers
      and 'government did have control over third party' is true in answers,
       if 'government did exercise adequate control' is true in answers,
      send {2 lines,"The government did exercise adequate control over",cr,
                "the actions of the third party.",cr},
        otherwise if 'government did not exercise adequate control'
                                              is true in answers,
            do nothing,
        otherwise if 'if-government did exercise adequate control'
                                              is true in unknowns,
      go unknown-adequate-control,
        otherwise,
      go determine-adequate-control.

[8] If 'government did not exercise adequate control' is true in answers,
            send {2 lines,"The government failed to exercise adequate",cr,
                     "control over the actions of the third ",cr,
                     "party.  Because of this the condition ",cr,
                     "may be ruled to be an exception to the ",cr,
                     "to the after-bid exclusion and may be",cr,
                     "within the scope of clause.",cr}
            and go add 'condition is possible exception-to
                           after-bid exclusions'
                               to reports
            and return.

[9] If 'condition did result from act-of-god' is true in answers
       or 'condition did result from no-one' is true in answers
        or 'condition did result from no-one' is true in assumptions
         or ('condition did result from act-of-third-party' is true in answers
        and ('government did not have control over third party'
                                   is true in answers
            or 'government did not have control over third party'
                                   is true in asssumptions
             or ('government did have control over third party'
                                   is true in answers
                and 'government did exercise adequate control'
                                   is true in answers
                   or 'government did exercise adequate control'
                                   is true in assumptions))),
      if 'condition did result from
                interacting-non-compensable-and-physical-factors'
                                              is true in answers,
        send {1 line,"The condition was caused by an interaction of",cr,
                "non-compensable conditions and physical factors",cr,
                "at the site, therefore, an exception may be ",cr,
                "allowed.",cr}
```

```
    and go add 'condition is possible
                  exception-to after-bid exclusions'
                                         to reports,
   otherwise if 'condition did not result from
                interacting-non-compensable-and-physical-factors'
                                         is true in answers,
    do nothing,
   otherwise if 'if-condition did result from
                interacting-non-compensable-and-physical-factors'
                                         is true in answers,
    go unknown-interaction-of-condition,
   otherwise,
    go determine-interaction-of-condition.

[10] If ('condition did not result from
                interacting-non-compensable-and-physical-factors'
                                         is true in answers
    or 'condition did not result from
                interacting-non-compensable-and-physical-factors'
                                     is true in assumptions),
  if 'condition did result from act-of-god' is true in answers,
    send {2 lines,"Acts of God are usually not considered to be",cr,
                "differing site conditions and exceptions are",cr,
                "not usually allowed unless an interaction ",cr,
                "with physical factors at the site occurred.",cr}
    and deny entitlement is still-probable
    and assert entitlement is not still-probable
    and return,
   otherwise if 'condition did result from act-of-third-party'
                                         is true in answers,
    go add 'man-made conditions are not usually allowed' to reports
    and send {2 lines,"Man made conditions usually are not allowed",cr,
                "as DSC's.  Especially when the cause was not ",cr,
                "under government control or was and the government",
           cr,"exercised adequate control or when no interaction",
           cr,"with physical factors at the site occurred.",cr}
    and deny entitlement is still-probable
    and assert entitlement is not still-probable
    and return.

[11] If 'condition did result from
                interacting-non-compensable-and-physical-factors'
                                         is true in answers,
    if 'condition did result from act-of-god' is true in answers
    or 'condition did result from act-of-third-party' is true in answers,
   go add 'condition is possible exception-to after-bid exclusions'
                                            to reports
   and send {2 lines,"Since an interaction with the physical factors",cr,
                "at the site did occur, the condition may be an",cr,
                "exception to the after-bid exclusions and may be",cr,
                "within the scope of the clause.",cr}
   and return.
```

```
[12] If 'condition is differing quantity' is true in answers,
       send {1 lines,"The condition is one of differing quantities.",cr},
         otherwise if 'condition is not differing quantity' is true in answers,
              send {1 lines,
              "The condition is also not one of differing quantities",cr}
              and return,
         otherwise if 'if-condition is differing quantity' is true in unknowns,
              go unknown-differing-quantities,
         otherwise,
              go determine-differing-quantities.

[13] If 'condition is not differing quantity' is true in answers
         or 'condition is not differing quantity' is true in assumptions,
        if 'condition did result from
                     interacting-non-compensable-and-physical-factors'
                                             is true in answers,
           go add 'condition is possible exception-to after-bid exclusions'
                                             to reports
           and return,
         otherwise,
           send {1 line,"Therefore, it is not considered a differing",cr,
                     "site condition and an exception probably will",cr,
                     "not be allowed.",cr}
           and deny entitlement is still-probable
           and assert entitlement is not still-probable
           and return.

[14] If 'condition is differing quantity' is true in answers,
        if 'contract does have var-in-est-quant-clause' is true in answers,
              send {3 lines,"Since the contract does have a variation in estimated",
                        cr,"quantity clause it is best to seek relief through",cr,
                           "this clause.",cr}
              and deny entitlement is still-probable
              and assert entitlement is not still-probable
              and return,
         otherwise if 'contract does not have var-in-est-quant-clause'
                                      is true in answers,
              send {1 lines,"The contract does not have a variation in estimated",cr,
                        "quantity clause.",cr},
         otherwise if 'if-contract does have var-in-est-quant-clause'
                                      is true in unknowns,
              go unknown-clause-for-differing-quantities,
         otherwise,
              go determine-clause-for-differing-quantities.

[15] If 'contract does not have var-in-est-quant-clause' is true in answers
         or 'contract does not have var-in-est-quant-clause'
                                          is true in assumptions,
       go add 'condition is possible exception-to after-bid exclusions'
                                                   to reports
       and send {2 lines,"However, since the condition is one of differing",
                 cr,"quantities it may be an exception to the after-",cr,
                    "bid exclusions and may be within the scope of",cr,
```

"the clause.",cr}.

End.

```
[: DRIVERS3 parsed Mon Dec  5 02:30:35 1983 by Kruppen :]

[*********************** EXPRESS/IMPLIED DEFICIENCIES **********************]

To check-express-implied-deficiencies:

[1] Assert express-implied-deficiencies was checked.

[2] If 'contract did contain statements concerning condition'
                                        is true in answers,
        send {3 lines,"The contract did contain statements concerning the",cr,
                  "relevant subsurface or latent conditions at ",cr,
              "the site.",cr}
        and go add 'contract did contain indications
                                    concerning condition' to reports,
    otherwise if 'contract did not contain statements concerning condition'
                                        is true in answers,
       do nothing,
    otherwise if 'if-contract did contain indications concerning
                                       condition' is true in unknowns,
        go unknown-express-conditions,
    otherwise,
        go determine-express-conditions.

[3] If ('contract did not contain statements concerning condition'
                                        is true in answers
      or 'contract did not contain statements concerning condition'
                                    is true in assumptions),
       if 'contract did contain general conditions implying condition'
                                        is true in answers,
          send {1 lines,"The general conditions of the contract implied ",cr,
              "the existence of the condition.",cr}
          and go add 'contract did contain indications concerning condition'
                                           to reports,
       otherwise if 'contract did not contain
                      general conditions implying condition'
                                       is true in answers,
          do nothing,
       otherwise if 'if-contract did contain
                      general conditions implying condition'
                                       is true in unknowns,
          go unknown-implied-conditions,
       otherwise,
          go determine-implied-conditions.

[4] If ('contract did not contain
              general conditions implying condition'
                                           is true in answers
       or 'contract did not contain general conditions implying condition'
                                    is true in assumptions),
       if 'contract did lead one-to infer conditions' is true in answers,
           send {1 lines,"Subsurface or latent conditions were ",cr,
```

```
          "inferred from the contract as a whole.",cr},
     otherwise if 'contract did not lead one-to infer conditions'
                                   is true in answers,
      do nothing,
     otherwise if 'if-contract did lead one-to infer conditions'
                                   is true in unknowns,
      go unknown-inferred-conditions,
     otherwise,
      go determine-inferred-conditions.

[5] If 'contract did lead one-to infer conditions' is true in answers,
   if 'contract information did justify inference' is true in answers,
    send {1 lines,"This inference was justified based on the information",
              cr,"in the contract.",cr}
        and go add 'contract did contain indications concerning condition'
                                      to reports,
    otherwise if 'contract information did not justify inference'
                                      is true in answers,
        send {1 lines,"This inference was not justified based on the ",cr,
                  "information in the contract.",cr},
    otherwise if 'if-contract information did justify inference'
                                      is true in unknowns,
        go unknown-if-inference-justified,
    otherwise,
        go determine-if-inference-justified.

[6] If (('contract did not contain statements concerning condition'
                                   is true in answers
     or 'contract did not contain statements concerning condition'
                               is true in assumptions)
     and ('contract did not contain general conditions implying condition'
                                      is true in answers
       or 'contract did not contain general conditions implying condition'
                                      is true in assumptions)
     and (('contract did not lead one-to infer conditions'
                                   is true in answers
         or 'contract did not lead one-to infer conditions'
                               is true in assumptions)
          or ('contract information did not justify inference'
                                   is true in answers
        or 'contract information did not justify inference'
                               is true in assumptions))),
        send {3 lines,"There were no indications regarding the condition",cr,
              "in the contract, therefore, a Type II condition",cr,
              "is the most probable avenue of recovery.",2 lines}
        and go add 'contract did not contain indications concerning condition'
                                      to reports.

[7] If 'contract did contain indications concerning condition'
                                   is true in reports,
   send {3 line,"There were indications regarding the condition ",cr,
        "in the contract, therefore, a Type I condition is ",cr,
             "the most probable avenue of recovery.",2 lines}.
```

```
End.

[***************** LATENT DEFICIENCIES/CONTRACT INDICATIONS ****************]

To check-latent-deficiencies:

[1] Assert latent-deficiencies was checked.

[2] If 'contract indications did contain latent deficiencies'
                                    is true in answers,
      send {3 lines,"There were latent deficiencies in the contract",cr,
                    "indications.",cr},
    otherwise if 'contract indications did not contain latent deficiencies'
                                                is true in answers,
      send {3 lines,"There were not latent deficiencies in the contract",cr,
                    "indications.",cr}
      and deny entitlement is still-probable
      and assert entitlement is not still-probable
      and return,
    otherwise if 'if-contract indications did contain latent deficiencies'
                                          is true in unknowns,
      go unknown-latent-deficiencies,
    otherwise,
      go determine-latent-deficiencies.

[3] Go check-lat-ind-common-info.

End.

To check-contract-indications:

[1] Assert contract-indications was checked.

[2] Go check-lat-ind-common-info.

End.

To check-lat-ind-common-info:

[1] If contract-indications was checked
      or ('contract indications did contain latent deficiencies'
                                    is true in answers
          or 'contract indications did contain latent deficiencies'
                                    is true in assumptions),
      if 'contract information did contain physical data' is true in answers,
         send {2 lines,
               "The physical data was made part of the contract.",cr},
       otherwise if 'contract information did not contain physical data'
                                    is true in answers,
        send {2 lines,"The physical data was not made part of the contract."
```

```
                                    ,cr},
        otherwise if 'if-contract information did contain physical data'
                                    is true in unknowns,
            go unknown-physical-data,
        otherwise,
            go determine-physical-data.

[2] If 'contract information did contain physical data' is true in answers,
        if 'contract information did reveal deficiencies/conditions'
                                    is true in answers,
          deny entitlement is still-probable
          and assert entitlement is not still-probable
          and (if latent-deficiencies was checked,
                send {1 line,"The physical data contained in the ",cr,
                "contract would have revealed or resolved the ",cr,
                "deficiencies in the contract indications.",cr},
           otherwise send {1 line,"The physical data contained in the",cr,
                     "contract would have indicated the nature",cr,
                     "of the existing conditions.",cr})
          and return,
        otherwise if 'contract information did not
                     reveal deficiencies/conditions'
                                    is true in answers,
          go add 'contractor is not required-to inspect other data'
                                    to reports
          and (if latent-deficiencies was checked,
            send {1 line,"The physical data contained in the contract",cr,
                "would not have revealed or resolved the ",cr,
                "deficiencies in the contract indications, ",cr,
                "therefore, the contractor is not bound to ",cr,
                "the requirement to inspect the other data.",cr},
           otherwise send {1 line,"The physical data contained in the",cr,
                     "contract would not ha'e indicated the ",cr,
                     "nature of the existing condition.",cr})
            and go add 'contractor is not required-to inspect other data'
                                    to reports
            and return,
        otherwise if 'if-contract information did
                     reveal deficiencies/conditions'
                                    is true in unknowns,
            go unknown-value-of-other-data,
        otherwise,
            go determine-value-of-other-data.

End.

[************************** STANDARD OF COMPARISON **************************]

To check-standard-of-comparison:

[1] Assert standard-of-comparison was checked.
```

```
[2] If 'expected conditions are establishable from customs-of-trade'
                                    is true in answers,
      send {3 lines,"The contractor can establish expected conditions ",cr,
                  "from customs of the trade.",cr},
    otherwise if 'expected conditions are establishable
                                from general-knowledge-in-industry'
                                    is true in answers,
      send {3 lines,"The contractor can establish expected conditions",cr,
                  "from general knowledge in the industry.",cr},
    otherwise if 'expected conditions are establishable
                                from manufacturers-recommendations'
                                    is true in answers,
      send {3 lines,"The contractor can establish expected conditions",cr,
                  "from manufacturers recommendations.",cr},
    otherwise if 'expected conditions are not establishable'
                                    is true in answers,
      send {3 lines,"The contractor cannot establish the expected",cr,
                  "conditions.",cr}
      and deny entitlement is still-probable
      and assert entitlement is not still-probable
      and return,
    otherwise if 'if-expected conditions are establishable'
                                    is true in unknowns,
      go unknown-establishment-of-standard,
    otherwise,
      go determine-establishment-of-standard.

[3] If ('expected conditions are establishable from customs-of-trade'
                                    is true in answers
        or 'expected conditions are establishable
                                from general-knowledge-in-industry'
                                    is true in answers
        or 'expected conditions are establishable
                                from manufacturers-recommendations'
                                    is true in answers),
      go add 'contractor is able-to establish standard-of-expectation'
                                    to reports.

[4] If 'contractor is able-to establish standard-of-expectation'
                                    is true in reports,
      if 'condition does differ from standard-of-expectation'
                                    is true in answers,
            send {1 lines,"The actual condition does differ from the",cr,
                        "standard of expectation.",cr}
            and return,
       otherwise if 'condition does not differ from standard-of-expectation'
                                    is true in answers,
            send {1 lines,"The actual condition does not differ from",
                        cr,"the standard of expectation.",cr}
            and deny entitlement is still-probable
            and assert entitlement is not still-probable
            and return,
       otherwise if 'if-condition does differ from standard-of-expectation'
```

```
                    is true in unknowns,
            go unknown-different-from-standard,
        otherwise,
            go determine-different-from-standard.

End.

[************************ SUPERIOR KNOWLEDGE ***************************]

To check-superior-knowledge-I:

[1] Assert knowledge-I was checked
            and assert contract-deficiency is a knowledge
            and assert contract-indications is a bit-o-info.

[2] Go check-knowledge-common-info.

End.

To check-superior-knowledge-II:

[1] Assert knowledge-II was checked
            and assert actual-conditions is a knowledge
            and assert standard-of-expectation is a bit-o-info.

[2] Go check-knowledge-common-info.

End.

To check-knowledge-common-info:

[1] Assert government is a party-in-question
      and if ('government was aware-of contract-deficiency'
                              is true in answers
            or 'government was aware-of actual-conditions'
                                  is true in answers),
            send {1 lines,"The government was aware of the ",
                    the knowledge,".",cr}
            and forget about party-in-question,
        otherwise if ('government was not aware-of contract-deficiency'
                              is true in answers
                  or 'government was not aware-of actual-conditions'
                              is true in answers),
            send {1 lines,"The government was not aware of the ",
                    the knowledge,".",cr}
            and forget about party-in-question,
        otherwise if ('if-government was aware-of contract-deficiency'
                              is true in unknowns
                  or 'if-government was aware-of actual-conditions'
                              is true in unknowns),
```

```
          go unknown-knowledge,
      otherwise,
          go determine-knowledge.

[2] If ('government was aware-of contract-deficiency' is true in answers
          or 'government was aware-of actual-conditions'
                                        is true in answers),
      if ('government did reveal contract-deficiency' is true in answers
          or 'government did reveal actual-conditions'
                                        is true in answers),
          send {1 lines,"The government did communicate its knowledge of the",
                    the knowledge," to the contractor.",cr},
      otherwise if ('government did not reveal contract-deficiency'
                                        is true in answers
                    or 'government did not reveal actual-conditions'
                                        is true in answers),
          send {3 lines,"The government did not communicate its knowledge ",cr,
              "of the ",the knowledge," to the contractor.",cr,
              "Possible breach by misrepresentation has occurred",cr,
              "however, I will continue our analysis.",cr}
          and go add 'government did have superior knowledge' to reports,
      otherwise if ('if-government did reveal contract-deficiency'
                                        is true in unknowns
                    or 'if-government did reveal actual-conditions'
                                        is true in unknowns),
          go unknown-communication-of-know,
      otherwise,
          go determine-communication-of-knowledge.

[3] If ('government did reveal contract-deficiency' is true in answers
        or 'government did reveal actual-conditions' is true in answers),
      if 'proof does indicate government-communicated-knowledge'
                                        is true in answers,
          do nothing,
      otherwise if 'proof does not indicate
                    government-communicated-knowledge'
                                        is true in answers,
          send {1 lines,"Assertions that the government notified the ",cr,
                    "contractor probably cannot be proven.",cr},
      otherwise if 'if-proof does indicate government-communicated-knowledge'
                                        is true in unknowns,
          assert government-communicated-knowledge is a statement
          and go unknown-evidence,
      otherwise,
          assert government-communicated-knowledge is a statement
          and go determine-evidence.

[4] If 'proof does indicate government-communicated-knowledge'
                                        is true in answers,
          send {1 lines,"There is evidence to prove that the government ",cr,
              "did reveal the ",the knowledge," to the contractor.",cr}
          and deny entitlement is still-probable
          and assert entitlement is not still-probable
```

```
        and return.

[5] Assert contractor is a party-in-question
      and if ('contractor was aware-of contract-deficiency'
                                          is true in answers
            or 'contractor was aware-of actual-conditions'
                                          is true in answers),
         send {3 lines,"The contractor was aware of the ",the knowledge,".",cr}
         and forget about the party-in-question,
       otherwise if ('contractor was not aware-of contract-deficiency'
                                          is true in answers
                    or 'contractor was not aware-of actual-conditions'
                                          is true in answers),
         send {3 lines,"The contractor was not aware of the ",
                                          the knowledge,".",cr},
       otherwise if ('if-contractor was aware-of contract-deficiency'
                                          is true in unknowns
                    or 'if-contractor was aware-of actual-conditions'
                                          is true in unknowns),
         go unknown-knowledge,
       otherwise,
         go determine-knowledge.

[6] If ('contractor was aware-of contract-deficiency'
                                          is true in answers
       or 'contractor was aware-of actual-conditions'
                                          is true in answers),
      if 'proof does indicate contractor-had-knowledge'
                                          is true in answers,
            do nothing,
       otherwise if 'proof does not indicate contractor-had-knowledge'
                                          is true in answers,
            send {1 lines,"Assertions of contractor's superior knowledge ",
                         "probably can't be proven.",cr},
       otherwise if 'if-proof does indicate contractor-had-knowledge'
                                          is true in unknowns,
            assert contractor-had-knowledge is a statement
            and go unknown-evidence,
       otherwise,
            assert contractor-had-knowledge is a statement
            and go determine-evidence.

[7] If 'proof does indicate contractor-had-knowledge' is true in answers,
       send {1 lines,"Proof of contractor's superior knowledge does exist.",
                                                      cr}
       and deny entitlement is still-probable
       and assert entitlement is not still-probable
       and return.

[8] If 'simple inquiry did give potential to-know contrary conditions'
                                          is true in answers,
       send {3 lines,"It has become obvious that the contractor did",cr,
                 "not simple inquiries.",cr,
```

```
                    "These simple inquiries by the contractor would",cr,
                    "have revealed the condition.",cr}
     and deny entitlement is still-probable
     and assert entitlement is not still-probable
     and return,
   otherwise if 'simple inquiry did not give potential
                              to-know contrary conditions'
                                           is true in answers,
     send {1 lines,"Simple inquiries by the contractor would not",cr,
                    "have revealed conditions to be contrary to those ",cr,
                    "indicated.",cr}
     and return,
   otherwise if 'if-simple inquiry did give potential
                              to-know contrary conditions'
                                           is true in unknowns,
     go unknown-effect-of-simple-inquiry,
   otherwise,
     go determine-effect-of-simple-inquiry.

End.

[**************************** SITE INVESTIGATION ***************************]

To check-site-inspection:

[1] Assert site-inspection was checked.

[2] If 'government did withhold access-to-site' is true in answers,
      send {3 lines,"The government denied access to the site.",cr}
      and go add 'site-inspection is not required' to reports
      and return,
    otherwise if 'government did allow inadequate time' is true in answers,
      send {3 lines,"The government allowed inadequate time for ",cr,
                    "site inspection.",cr}
      and go add 'site-inspection is not required' to reports
      and return,
    otherwise if 'government did not hinder inspection' is true in answers,
      send {3 lines,"The government did not hinder the site inspection ",cr,
                    "in any way.",cr},
    otherwise if 'if-government did hinder inspection' is true in unknowns,
      go unknown-hinderance-of-inspection,
    otherwise,
      go determine-hinderance-of-inspection.

[3] If ('government did not hinder inspection' is true in answers
        or 'government did not hinder inspection' is true in assumptions),
      if 'contractor did conduct site-inspection' is true in answers,
      send {1 lines "The contractor did conduct a site inspection.",cr},
      otherwise if 'contractor did not conduct site-inspection'
                              is true in answers,
          send {1 lines,"The contractor failed to conduct a site inspection.",
          "He risked encountering an unexpected condition",cr,
```

```
            "that a reasonable inspection might have revealed.",cr}
            and go add 'contractor did not conduct site-inspection'
                                to reports,
        otherwise if 'if-contractor did conduct site-inspection'
                                is true in unknowns,
            go unknown-if-inspection-made,
        otherwise,
            go determine-if-inspection-made.

[4] If 'contractor did conduct site-inspection' is true in answers,
        if 'site-inspection was reasonable inspection' is true in answers,
          send {3 lines,"Inspection conducted by the contractor was a ",cr,
                  "reasonable site inspection.",cr}
          and return,
        otherwise if 'site-inspection was not reasonable inspection'
                                      is true in answers,
          send {3 lines,"Inspection conducted by the contractor was not a",cr,
              "reasonable site inspection.",cr},
        otherwise if 'if-site-inspection was reasonable inspection'
                                      is true in unknowns,
          go unknown-reasonable-inspection,
        otherwise,
          go determine-reasonable-inspection.

[5] If 'reasonable inspection did not have potential-to-reveal condition'
                                     is true in answers,
        send {3 lines,"Actual conditions would not have been discernable ",cr,
            "from a reasonable inspection, therefore, reasonable ",cr,
            "site inspection is not required for entitlement.",cr}
        and go add 'reasonable inspection is not required-for-entitlement'
                                          to reports,
      otherwise if 'reasonable inspection did have
                          potential-to-reveal condition'
                                is true in answers,
        do nothing,
      otherwise if 'if-reasonable inspection did have
                          potential-to-reveal condition'
                                is true in unknowns,
        go unknown-unreasonable-inspection,
      otherwise,
        go determine-unreasonable-inspection.

[6] If 'reasonable inspection did have potential-to-reveal condition'
                                     is true in answers,
        send {3 lines,"Actual conditions would have been discernable ",cr,
                  "from a reasonable inspection.",cr}
        and deny entitlement is still-probable
        and assert entitlement is not still-probable
        and if ('site-inspection was reasonable inspection' is true in answers
         or 'site-inspection was reasonable inspection'
                                     is true in assumptions),
          send {2 lines,"Contractor's error indicates that he did not",cr,
                  "conduct a reasonable inspection.",cr}
```

```
        and go add 'site-inspection was unreasonable' to reports.

End.

[****************************** RELIANCE ******************************]

To check-reliance-I:

[1] Assert reliance-I was checked
    and assert deficient-indications-in-contract is a statement.

[2] Go check-reliance-common-info.

[3] If 'contractor did not suffer prejudice through reliance' is true in answers,
      send {3 lines,"The difference between actual and expected",cr,
            "conditions did not prejudice the contractor",cr,
            "since his bid did not reflect the indicated",cr,
            "condition.",cr}
         and go add 'contractor bid did not reflect condition' to reports
         and assert entitlement is not still-probable.

End.

To check-reliance-II:

[1] Assert reliance-II was checked
    and assert expectations-that-condition-would-meet-standard is a statement.

[2] Go check-reliance-common-info.

[3] If 'contractor did not suffer prejudice through reliance' is true in answers,
      send {3 lines,"The difference between actual and expected",cr,
            "conditions did not prejudice the contractor",cr,
            "since his bid did not reflect the condition.",cr}
         and deny entitlement is still-probable
         and assert entitlement is not still-probable.

End.

To check-reliance-common-info:

[1] If 'contractor did suffer prejudice through reliance' is true in answers,
      send {3 lines,"The contractor did suffer prejudice through his",cr,
            "reliance on the ",the statement,".",cr},
    otherwise if 'contractor did not suffer prejudice through reliance'
                             is true in answers,
      do nothing,
    otherwise if 'if-contractor did suffer prejudice through reliance'
                             is true unknowns,
      go unknown-reliance,
```

```
    otherwise,
      go determine-reliance.

[2] Forget about the statement.

End.


[*************************** MATERIAL DIFFERENCE *****************************]

To check-material-difference-I:

[1] Assert material-difference-I was checked
      and assert indicated is a statement
      and assert indicated is a comment.

[2] Go check-material-difference-common-info.

[3] If 'difference was material' is true in answers,
      send {2 lines,"Relief is probably available through the Type I ",cr,
            "DSC clause because of deficient  indications in ",cr,
            "the contract.",cr}.

[4] Forget about the statement.

End.


To check-material-difference-II:

[1] Assert material-difference-II was checked
      and assert expected is a statement
      and assert recognized-and-usual is a comment.

[2] Go check-material-difference-common-info.

[3] If 'difference was material' is true in answers,
      send {2 lines,"Relief is probably available through the Type II",cr,
            "DSC clause because of unanticipated or unusual ",cr,
            "conditions or materials.",cr}.

[4] Forget about the statement.

End.


To check-material-difference-common-info:

[1] If 'difference was material' is true in answers,
      send {3 lines,"The difference between actual and ",the statement,cr,
                    " conditions is material.",cr}
      and return,
    otherwise if 'difference was not material' is true in answers,
```

```
      send {3 lines,"The difference between actual and ",the statement,cr,
           " conditions is not material. The contractor ",cr,
           "should have anticipated the condition as",
           the comment,".",cr}
      and deny entitlement is still-probable
      and assert entitlement is not still-probable
      and return,
    otherwise if 'if-difference was material' is true in unknowns,
      go unknown-material-difference,
    otherwise,
      go determine-material-difference.

End.

[************************* EXCULPATORY LANGUAGE ***************************]

To check-exculpatory-language-I:

[1] Assert exculpatory-language-I was checked.

[2] If 'contract does contain exculpatory-clause' is true in answers,
      send {3 lines,"The contract contains an exculpatory clause",cr,
           "denying any government liability for actual",cr,
           "conditions different from those indicated in the",cr,
           "contract documents.",cr},
    otherwise if 'contract does not contain exculpatory-clause'
                                    is true in answers,
      send {3 lines,"The contract does not contain an exculpatory",
                cr,"clause.",cr,
                "Therefore, entitlement will probably be allowed.",cr,
                "Consult your local CO for further details.",cr}
      and return,
    otherwise if 'if-contract does contain exculpatory-clause'
                                    is true in unknowns,
      go unknown-exculpatory-clause-exists,
    otherwise,
      go determine-exculpatory-clause-exists.

[3] Go check-exculpatory-language-common-info.

End.


To check-exculpatory-language-II:

[1] Assert exculpatory-language-II was checked.

[2] If 'contract does contain exculpatory-clause' is true in answers,
      send {3 lines,"The contract contains a clause placing the risk",cr,
                "of conditions not revealed by the contractor's",cr,
                "site investigation.",cr},
    otherwise if 'contract does not contain exculpatory-clause'
```

```
                    is true in answers,
   send {3 lines,"The contract does not contain a site inspection",cr,
          "exculpatory clause.",cr,
          "Therefore, entitlement will probably be allowed.",cr,
          "Consult your local CO for further details.",cr}
   and return,
  otherwise if 'if-contract does contain exculpatory-clause'
                    is true in unknowns,
   go unknown-exculpatory-clause-exists,
  otherwise,
   go determine-exculpatory-clause-exists.

[3] Go check-exculpatory-language-common-info.

End.


To check-exculpatory-language-common-info:

[1] If 'contract does not contain exculpatory-clause' is true in assumptions,
     send {1 line, "I have assumed that the contract does not contain",cr,
           "an exculpatory clause.  Therefore, entitlement will",cr,
           "probably be allowed.",2 lines,
           "Consult your local CO for further details.",cr}
      and return.

[2] If 'contract does contain exculpatory-clause' is true in answers,
     if 'exculpatory-clause is specific-to DSC clause' is true in answers,
          send {1 line,"The exculpatory clause is specific to the",cr,
                "DSC clause.",cr},
      otherwise if 'exculpatory-clause is not specific-to DSC clause'
                    is true in answers,
          send {1 line,"The exculpatory clause is not specific to",cr,
                "the DSC clause.",cr}
        and go add 'validity-of exculpatory-clause is in_doubt' to reports,
      otherwise if 'if-exculpatory-clause is specific-to DSC clause'
                    is true in unknowns,
          go unknown-specific-to-clause,
      otherwise,
          go determine-specific-to-clause.

[3] If 'exculpatory-clause is specific-to DSC clause' is true in answers
      or ('exculpatory-clause is not specific-to DSC clause'
                    is true in answers
       or 'exculpatory-clause is not specific-to DSC clause'
                    is true in assumptions),
     if 'exculpatory-clause is clear-and-unambiguous'
                    is true in answers,
          send {1 line,"The exculpatory language is clear and unambiguous.",
                     1 line},
      otherwise if 'exculpatory-clause is not clear-and-unambiguous'
                    is true in answers,
          send {1 line,"The exculpatory language is not",cr,
```

```
            "clear and unambiguous.",1 line},
    otherwise if 'if-exculpatory-clause is clear-and-unambiguous'
                          is true in unknowns,
        go unknown-clarity-of-clause,
    otherwise,
        go determine-clarity-of-clause.

[4] If ('exculpatory-clause is specific-to DSC clause' is true in answers
    or 'exculpatory-clause is specific-to DSC clause'
                              is true in assumptions),
    if ('exculpatory-clause is clear-and-unambiguous'
                              is true in answers
      or 'exculpatory-clause is clear-and-unambiguous'
                              is true in assumptions),
        send {2 lines,"I believe that the excupatory clause ",cr,
                 "is probably valid for this case.",cr}
        and deny entitlement is still-probable
        and assert entitlement is not still-probable
        and return,
    otherwise if 'exculpatory-clause is not clear-and-unambiguous'
                              is true in answers,
        send {2 lines,"Since the exculpatory language is specific",cr,
               "to the DSC clause there is a 50/50 chance ",cr,
               "that entitlement is justified.",2 lines,
               "Consult your local CO for further details.",cr}
        and return.

[5] If 'exculpatory-clause is not specific-to DSC clause'
                              is true in answers,
    if ('exculpatory-clause is clear-and-unambiguous'
                              is true in answers
       or 'exculpatory-clause is clear-and-unambiguous'
                              is true in assumptions),
        send {2 clause,"Since the exculpatory language is not",cr,
        "specific to the DSC clause there is only a",cr,
        "50/50 chance that entitlement will be justified.",2 lines,
        "Consult your local CO for further details.",cr}
        and return,
    otherwise if 'exculpatory-clause is not clear-and-unambiguous'
                              is true in answers,
        send {2 lines,"I believe that the exculpatory clause is",cr,
            "probably not valid for this case.",3 lines,
            "Therefore, entitlement will probably be allowed.",cr,
            "Consult your local CO for further details.",cr}
        and go add 'exculpatory-clause is not probably valid'
                                  to reports.

End.
```

[: DSCASDATA parsed Wed Nov 30 12:07:46 1983 by Kruppen :]

[rule 1] Assert DSCAS is running.

[rule 2] Assert cause-of-condition is unknown.

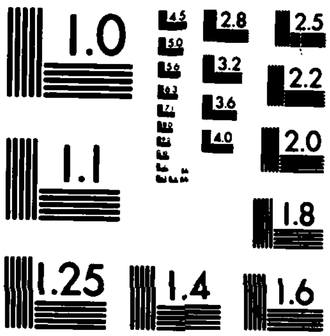
[rule 3] Assert entitlement is still-probable.

[rule 4] Let the ucount be 0.

[rule 5] Let the instance of 'the qnum' be 1.

AD-A146 681 THE APPLICATION OF ARTIFICIAL INTELLIGENCE TO CONTRACT MANAGEMENT(U) CONSTRUCTION ENGINEERING RESEARCH LAB (ARMY) CHAMPAIGN IL T A KRUPPENBACHER AUG 84 4/6

UNCLASSIFIED CERL-TM-P-166 F/G 9/1 NL



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

```
[: ENTITLE parsed Tue Nov 29 09:49:03 1983 by Kruppen :]

[************************ DETERMINE ENTITLEMENT ****************************]

To check-entitlement:

[1] Send {2 lines,"Thinking.......",2 lines}
        and activate reports.

[2] If contractor did probably concede rights-to-claim
        or contract did assign obligation-risk
         or condition is not considered to-be DSC
          or contractor did contribute to condition
           or condition is excluded condition
            or condition is best claimed through var-in-est-quant-clause
             or contractor did fail to-heed contract indications
              or contractor is not able-to establish standard-of-expectation
               or condition does not differ from standard-of-expectation
                or contractor did have imputed knowledge-of-condition
                 or contractor did have superior knowledge
                  or contractor did not make simple inquiries
                   or reasonable inspection is required-for-entitlement
                    or contractor bid did not reflect condition
                     or difference was not material
                      or exculpatory-clause is probably valid
        assert reason_for no entitlement does exist.

[3] If reason_for no entitlement does exist,
            send {2 lines,
              "**** Entitlement will probably not",1 blank,
                   "be allowed under DSC claim. ****",3 lines}
            and send {"I believe that entitlement will probably",1 blank,
                   "not be allowed because I",cr,
                   "concluded that "}
            and open entitle.text to read
            and go explain-entitlement
            and deactivate,
         otherwise,
            deny entitlement is not still-probable
            and assert entitlement is still-probable
            and deactivate.

[4] If exculpatory-language-I was checked
        or exculpatory-language-II was checked,
            assert analysis is complete.

End.
```

```
[: EXP-QUESTS parsed Mon Dec  5 00:36:05 1983 by Kruppen :]

To explain-quests:

[1] Choose situation:

  If dfon is a question,
    send {2 lines,"Your choices for this question include:",2 lines,
    "   a. Written notice was given -- ",cr,
    "         Self explanatory.",2 lines,
    "   b. Oral notice was given -- ",cr,
    "         Self explanantory.",2 lines,
    "   c. Other events surrounded notice --",cr,
    "         You should select this answer if the government",cr,
    "         became aware of the condition through alternate",cr,
    "         means or if the notice requirement was not waived",cr,
    "         for some reason.",2 lines,
    "Note: The selection of choice 'c' will lead you to other ",cr,
    "     questions designed to elaborate on the other possible",cr,
    "     events which may have occurred regarding government",cr,
    "     notification of the condition.",2 lines,": "};
  If don is a question,
    send {2 lines,"Your choices for this question are as follows:",2 lines,
    "   a. An emergency situation was encountered requiring ",cr,
    "      the contractor to act immediately. --",cr,
    "         This answer should be selected if the contractor",cr,
    "         was unable to give notice before disturbing the",cr,
    "         condition or before performing remedial work ",cr,
    "         because the situation encountered presented an",cr,
    "         immenent disaster if neglected.",2 lines,
    "   b. Notice of the occurrence of the same condition had been",cr,
    "      previously received by the government for this job. --",cr,
    "         This answer should be selected if the government was",cr,
    "         was made aware of the condition previously by other",cr,
    "         parties working at the site.",2 lines,
    "   c. Other circumstances led to the waiver. --",cr,
    "         This answer should be selected if the notice",cr,
    "         requirment was waived due to some special",cr,
    "         circumstances, i.e., CO at the job waived",cr,
    "         requirement.  You will be asked to tell me the",cr,
    "         circumstances which you have in mind so that I",cr,
    "         might keep a record of them.",2 lines,
    "   d. Notice requirement was not waived. -- ",cr,
    "         Self-explanatory.",2 lines,": "};
  If doe is a question,
    send {2 lines,"For this question you are to specify the circumstances",cr,
          "which led you to beleive that the notice requirement had",cr,
          "been waived.  Your answer can be anything you wish,",cr,
          "however, it should be as accurate as possible.  I ",cr,
          "currently have no way of checking its validity so ",cr,
          "you're on your honor on this one.",2 lines,
          "  I prefer that you keep your answer to as few words as ",cr,
          "humanly possible, it is easiest for me to remember",cr,
```

```
                "responses of 4 to 6 words, so think about your response",cr,
                "before giving it to me.",2 lines};
If dhs-1 is a question,
  send {2 lines,"The choices for this question are as follows:",2 lines,
  "   a. Notice given before contractor disturbed condition. -- ",cr,
  "         Select if the contractor did not disturb the",cr,
  "         condition between the time the condition was",cr,
  "         discovered and the time notice was given.  Even",cr,
  "         if he performed some remedial work concerning the",cr,
  "         condition which may not have disturbed the",cr,
  "         condition.",2 lines,
  "   b. Notice given before contractor performed remedial work. -- ",
                      cr,
  "         Select if the contractor has not yet done any work which",cr,
  "         is related to the condition even though he may have ",cr,
  "         disturbed the condition.",2 lines,
  "   c. Both a and b -- ",cr,
  "         Select if the contractor has not disturbed the condition",cr,
  "         yet and has not performed any remedial work",cr,
  "         concerning the condition yet.",2 lines,
  "   d. None of the above -- ",cr,
  "         Self-explanatory.",2 lines,":  "};
If drr is a question,
  send {2 lines,"You have the following choices for this question:",2 lines,
  "   a. The contracting officer -- ",cr,
  "         Notice was given directly to the contracting officer ",cr,
  "         (CO) at the job.",2 lines,
  "   b. An authorized representative of the CO -- ",cr,
  "         Notice was given to a person who was authorized by the ",cr,
  "         CO to act on his behalf.",2 lines,
  "   c. Any other employee of the government -- ",cr,
  "         Notice was given to some employee of the government who",cr,
  "         was not the CO or a person who had been specifically",cr,
  "         authorized to be a a representative of the government",cr,
  "         at the job site, i.e., a secretary or someone in a ",cr,
  "         similar position.",2 lines,
  "   d. None of the above -- ",cr,
  "         Self-explanatory.  If you select this answer I am forced",cr,
  "         to conclude that the contractor's notification of the",cr,
  "         government is in doubt.",2 lines,":  "};
If do-1 is a question,
  send {2 lines,"You have the following choices for this question:",2 lines,
  "   a. The government -- ",cr,
  "         Select if the contract expressly assigns obligation for",cr,
  "         or risk of the encountered condition to the ",cr,
  "         government ONLY.",2 lines,
  "   b. The contractor -- ",cr,
  "         Select if the contract expressly assigns obligation for",cr,
  "         or risk of the encountered condition to the",cr,
  "         contractor ONLY.",2 lines,
  "   c. Both the government and the contractor -- ",cr,
  "         Select if the contract expressly assigns obligation for",cr,
  "         or risk of the encountered condition to BOTH the ",cr,
```

```
"           contractor and the government.  If you select this I am ",cr,
"           forced to conclude that there is confusion in the ",cr,
"           contract as to who has been assigned the risk of the",cr,
"           condition, however, I will continue our analysis.",2 lines,
"     d. Neither the government nor the contractor -- ",cr,
"           Select if the contract is SILENT regarding the",cr,
"           assignment of obligation for or risk of the condition.",
                  2 lines,": "};
If dco is a question,
  send {2 lines,"You may answer this question in the following manner:",
                          2 lines,
  "   Before -- If the condition occurred before the contract was",cr,
  "             awarded.",2 lines,
  "   After  -- If the condition occurred after the contract was",cr,
  "             awarded.",2 lines,
  "   Also you may answer it with a '-' if the answer is unknown.",
                  2 lines};
If dc is a question,
  send {2 lines,"Your choices for this question are:",2 lines,
  "Act-of-God ",cr,
  "Act-of-third-party",cr,
  "Act-of-government",cr,
  "Act-of-contractor",cr,
  "Fault-of-contractor",cr,
  "Fault-of-government",cr,
  "No-one",2 lines,
  "Please follow the format shown and type 'end.' on the line below ",cr,
  "your last answer to the question.  Also, remember to wait for the",cr,
  "prompt ':' before typing in each answer.",2 lines};
If deos is a question,
  send {2 lines,"The choices for this question are as follows:",2 lines,
  "   a. Customs of the trade -- ",cr,
  "         Select if the contractor can establish expected",cr,
  "         conditions based on what is customary for the trade.",2 lines,
  "   b. General knowledge in the industry -- ",cr,
  "         Select if the contractor can establish expected",cr,
  "         conditions based on information which is generally",cr,
  "         accepted knowledge in the industry.",2 lines,
  "   c. Manufacturers instructions or recommendations --",cr,
  "         Select if the contractor can establish expected",cr,
  "         conditions based on information from the",cr,
  "         manufacturers of the product.",2 lines,
  "   d. None of the above -- ",cr,
  "         Self-explanatory.  However, I must add that I will be",cr,
  "         forced to conclude that the contractor is unable to",cr,
  "         establish what expected conditions are.  This will",cr,
  "         result in no entitlement.",2 lines,": "};
If dhti is a question,
  send {2 lines,"The choices for this question are as follows:",2 lines,
  "   a. Access to site was denied -- ",cr,
  "         Select if the government denied the contractor access",cr,
  "         to the site for the purpose of conducting a site",cr,
  "         investigation prior to bidding.",2 lines,
```

```
"    b. Inadequate time was allowed -- ",cr,
"           Select if the government did not allow the contractor",cr,
"           enough time to perform an adequate site inspection prior",cr,
"           to bidding.",2 lines,
"    c. No act of the government hindered inspection -- ",cr,
"           Select if the the statement describes the situation",cr,
"           of the instant case or if none of the above choices",cr,
"           are applicable, i.e., the choices above are the only",cr,
"           means by which the government can nullify the site",cr,
"           inspection requirement.",2 lines,": "}.
```

End.

To explain-date:

```
[1] Send {2 lines,"The date of ",the event," should be entered",cr,
        "in the following manner:",2 lines,
        "     Month Day, Year    e.g. April 3, 1983",2 lines,
        "Now you try it...",2 lines}.
```

End.

```
[: QUESTS parsed Mon Dec  5 00:28:50 1983 by Kruppen :]

[************************** DSC assertion and date ***************************]

To determine-if-DSC-was-asserted:

[1] Send {2 lines,"[",the qnum,"]",2 blanks,
      "Has a differing site condition been asserted by one of",cr,
      "your contractors? (Yes/No)",cr,":  "}
      and assert didwa is a question
      and go input-yes-no
      and increment 'the qnum'.

[2] If the lowercase of the string = "yes",
      go add 'DSC was asserted' to answers
      and assert DSC-assertion is an event
      and go get-date
      and increment 'the qnum'
    otherwise if the lowercase of the string = "no",
      send {1 line,"I'm sorry, but I can't help you with your problems.",
      2 lines}
      and go add 'DSC was not asserted' to answers
      and return,
    otherwise if the string = "-",
      go add 'if-DSC was asserted' to unknowns
      and go add 'DSC was asserted' to assumptions.

End.

[*********************** Was final payment made, when? *******************]

To determine-if-final-payment-was-made:

[1] Send {3 lines,"[",the qnum,"]",2 blanks,
      "Has final payment been made? (Yes/No)",cr, ":",2 blanks}
     and assert difpwm is a question
     and go input-yes-no
     and increment 'the qnum'.

[2] If the lowercase of the string = "no",
     go add 'final-payment was not made' to answers
     and return,
    otherwise if the lowercase of the string = "yes",
     deny entitlement is still-probable
     and assert entitlement is not still-probable
     and go add 'final-payment was made' to answers,
    otherwise if the string = "-",
     go add 'if-final-payment was made' to unknowns
     and go add 'final-payment was not made' to assumptions
     and return.
```

```
[3] Assert final-payment is an event
      and go get-date
      and increment 'the qnum'.

End.

[************************* Decide form of notice ***************************]

To determine-form-of-notice:

[1] Send {3 lines,"[",the qnum,"]",2 blanks,
      "What form of notice was given to the government by",cr,
      "the contractor? (Select one of the following.)",1 line,
      "       a. Written notice was given.",cr,
      "       b. Oral notice was given. ",cr,
      "       c. Other events surrounded notice. ",cr,": "}
      and assert dfon is a question
      and let the nchoices be 3
      and go read-vrbl
      and increment 'the qnum'.

[2] If the vrbl = "a",
      go add 'form-of-notice was written' to answers
      and go add 'contractor did comply with written-notice-requirement'
                                      to reports
      and assert written-notice is an event
      and go get-date
      and increment 'the qnum'
      and return,
    otherwise if the vrbl = "b",
      go add 'form-of-notice was oral' to answers
      and go add 'contractor did constructively comply with notice-requirement'
                                             to reports
      and assert oral-notice is an event
      and go get-date
      and increment 'the qnum'
      and return,
    otherwise if the vrbl = "c",
      go add 'form-of-notice was  other' to answers
      and return,
    otherwise if the vrbl = "-",
      go add 'form-of-notice is unknown' to unknowns
      and go add 'form-of-notice was other' to assumptions
      and go add 'contractor did constructively
                  comply with notice-requirement'
                                             to reports.

End.

To determine-other-notice:
```

```
[1] Send {2 lines,"[",the qnum,"]",2 blanks,
      "Was the notice requirement waived due to any of the following ",cr,
      "reasons? (If so, please select the appropriate circumstances.)",
      1 line,
      "      a. An emergency situation was encountered in which ",cr,
      "            the contractor was required to act immediately.",cr,
      "      b. Notice of the occurrence of the same differing site ",cr,
      "            condition had been previously received by the ",cr,
      "            government for this job.",cr,
      "      c. Other circumstances led to the waiver.",cr,
      "      d. Notice requirement was not waived.",cr,": "}
      and assert don is a question
      and let the nchoices be 4
      and go read-vrbl
      and increment 'the qnum'.

[2] If the vrbl = "a",
       go add 'emergency-situation was encountered' to answers
       and go add 'notice-requirement was waived' to reports
       and return,
     otherwise if the vrbl = "b",
       go add 'same-situation was encountered previously' to answers
       and go add 'notice-requirement was waived' to reports
       and return,
     otherwise if the vrbl = "c",
       go add 'reason-for-notice-waiver does exist' to answers
       and assert the reason-for-notice-waiver is an item
       and go determine-other-events
       and increment 'the qnum'
       and forget about item
       and return,
     otherwise if the vrbl = "d",
       go add 'notice-requirement was not waived' to answers
       and go add 'notice-requirement was not waived' to reports
       and go add 'actual-occurrence-of-notice is an in_doubt' to reports
       and send {1 line,"The actual occurrence of the notice to the ",cr,
                  "government is in doubt.",cr},
     otherwise if the vrbl = "-",
       go add 'if-notice-requirement was waived' to unknowns
       and go add 'notice-requirement was not waived' to assumptions
       and go add 'actual-occurrence-of-notice is an in_doubt' to reports.

End.

To determine-other-events:

Execute cyclically.

[1] Send {2 lines,"[",the qnum,"]",2 blanks,
      "Briefly state the other event which led you to beleive",cr,
      "that the notice requirement was the item,",cr,
      "i.e., Requirement was waived by CO., (Please",cr,
```

```
  "keep your answer brief, preferably 4-6 words.)",cr,":  "}
 and increment 'the qnum'
 and read {0 or more characters (bind the string),cr}
 and if the string = "q",
  assert system is stopped
       and go wrap-up,
   otherwise if the string = "?",
       go explain-quests,
   otherwise,
       evaluate {"go add '<",the string,
                         "> is a ",the item,"' to answers."}.

End.

[************************ Determine Promptness *************************]

To determine-promptness:

[1] Send {2 line,"[",the qnum,"]",2 blanks,
      "When was notice given in relation to the progression ",cr,
      "of work? ",cr,
      "a.  Notice given before contractor disturbed the condition.",cr,
      "b.  Notice given before contractor performed remedial work.",cr,
      "c.  Both a and b.",cr,
      "d.  None of the above.",cr,":  "}
    and assert dhs-1 is a question
    and let the nchoices be 4
    and go read-vrbl
    and increment 'the qnum'.

[2] If the vrbl = "a",
     go add 'notice was given before contractor disturbed condition'
                                         to answers
     and go add 'promptness_of_notice is an in_doubt' to reports
     and return,
    otherwise if the vrbl = "b",
     go add 'notice was given before remedial work performed'
                                         to answers
     and go add 'promptness_of_notice is an in_doubt' to reports
     and return,
    otherwise if the vrbl = "c",
     go add 'notice was given before contractor disturbed condition'
                                         to answers
     and go add 'notice was given before remedial work performed'
                                         to answers
     and return,
    otherwise if the vrbl = "d",
     go add 'notice was not given before remedial work performed'
                                         to answers
     and go add 'promptness_of_notice is an in_doubt' to reports
     and return,
    otherwise if the vrbl = "-",
```

```
    go add 'if-notice was given before condition disturbed' to unknowns
    and go add 'notice was given before condition disturbed'
                                     to assumptions
    and go add 'notice was given before remedial work performed'
                                          to assumptions.

End.


To determine-potential-promptness:

[1] Send {2 line,"[",the qnum,"]",2 blanks,
     "Is there evidence that the contractor should have been",cr,
     "aware of the differing site condition sooner? (Yes/No)",cr,":  "}
    and assert dhp-1 is a question
    and go input-yes-no
    and increment 'the qnum'.

[2] If the lowercase of the string = "no",
    go add 'contractor did not have potential-to-know condition sooner'
                                          to answers
    and go add 'notice was prompt' to reports
    and return,
   otherwise if the lowercase of the string = "yes",
    go add 'contractor did have potential-to-know condition sooner'
                                          to answers
    and go add 'promptness_of_notice is an in_doubt' to reports
    and return,
   otherwise if the string = "-",
    go add 'if-contractor did have potential-to-know condition sooner'
                                          to unknowns
    and go add 'contractor did not have potential-to-know condition sooner'
                                          to assumptions
    and go add 'notice was prompt' to reports.

End.


[**************************** RESPONSIBLE RECEIVER ***************************]

To determine-responsible-receiver:

[1] Send {3 lines,"[",the qnum,"]",2 blanks,
     "To whom was notice of the differing site condition given? ",cr,
     "a.  The contracting officer ",cr,
     "b.  An authorized representative of the contracting officer ",cr,
     "c.  Any other employee of the government ",cr,
     "d.  None of the above",cr,":  "}
    and assert drr is a question
    and let the nchoices be 4
    and go read-vrbl
    and increment 'the qnum'.
```

```
[2] If the vrbl = "a",
      go add 'notice was received-by contracting officer' to answers
      and go add 'notice was given-to responsible receiver' to reports
      and go add 'government did receive notice-of-condition' to reports,
    otherwise if the vrbl = "b",
      go add 'notice was received-by authorized representative' to answers
      and go add 'notice was given-to responsible receiver' to reports,
    otherwise if the vrbl = "c",
      go add 'notice was received-by other government employee' to answers,
    otherwise if the vrbl = "d",
      go add 'notice was not received by government' to answers
      and go add 'receipt-of-notice is an in_doubt' to reports,
    otherwise if the vrbl = "-",
      go add 'to-whom was notice given' to unknowns
      and send {1 line,"Receipt of notice by responsible government",cr,
            "employee is in doubt.",cr}
      and go add 'notice was not received-by government' to assumptions
      and go add 'receipt-of-notice is an in_doubt' to reports.

End.


To determine-communication-of-notice:

[1] Send {2 lines,"[",the qnum,"]",2 blanks,
        "Did this other government employee communicate his awareness",cr,
            "of the DSC to the CO? (Yes/No)",cr,": "}
      and assert doe-1 is a question
      and go input-yes-no
      and increment 'the qnum'.

[2] If the lowercase of the string = "yes",
      go add 'notice was given-to responsible receiver' to reports
      and go add 'government did receive notice-of-condition' to reports
      and go add 'government employee did communicate DSC awareness to-CO'
                              to answers,
    otherwise if the lowercase of the string = "no",
      go add 'government employee did not communicate DSC awareness to-CO'
                                   to answers
      and return,
    otherwise if the string = "-",
      go add 'if-government employee did communicate DSC awareness to-CO'
                                   to unknowns
      and go add 'government employee did
                 communicate DSC awareness to-CO'
                              to assumptions.

End.


To determine-position:

[1] Send {2 lines,"[",the qnum,"]",2 blanks,
```

```
        "Was this other government employee cognisant of the implications",cr,
        "of the condition? (Yes/No)",cr,": "}
       and assert dp is a question
       and go input-yes-no
       and increment 'the qnum'.

[2] If the lowercase of the string = "yes",
       go add 'government employee did understand implications' to answers
       and go add 'notice was given-to responsible receiver' to reports,
    otherwise if the lowercase of the string = "no",
       go add 'government employee did not understand implications' to answers
       and go add 'receipt-of-notice is an in_doubt' to reports
       and send {2 lines,"The person to whom notice was given is not",cr,
              "a responsible receiver of the notice, therefore",cr,
              "government receipt of notice is in doubt.",cr},
    otherwise if the string = "-",
       go add 'if-government employee did understand implications'
                                                  to unknowns
       and go add 'government employee did understand implications'
                                                  to assumptions
       and go add 'notice was given-to responsible receiver' to reports.

End.


To determine-proof:

[1] Send {2 lines,"[",the qnum,"]",2 blanks,
       "Does evidence exist to prove that notice was",cr,
       "given to the other government employee in question? (Yes/No)",
       cr,": "}
       and assert doe-2 is a question
       and go input-yes-no
       and increment 'the qnum'.

[2] If the lowercase of the string = "yes",
       go add 'proof-of DSC notice does exist' to answers
       and go add 'government was aware of DSC' to reports
       and return,
    otherwise if the lowercase of the string = "no",
       go add 'proof-of DSC notice does not exist' to answers
       and go add 'government awareness_of_DSC is an in_doubt' to reports
       and return,
    otherwise if the string = "-",
       go add 'if-proof-of DSC notice does exist' to unknowns
       and go add 'proof-of DSC notice does exist' to assumptions
       and go add 'government was aware of DSC' to reports.

End.
```

[******************************* PREJUDICE *********************************]

```
To determine-prejudice-from-passage-of-time:

[1] Send {3 lines,"[",the qnum,"]",2 blanks,
      "Is the government able to show that evidence against",cr,
      "the claim was obscurred due to the passage of time? (Yes/No)",
              cr,": "}
      and assert dpfpot is a question
      and go input-yes-no
      and increment 'the qnum'.

[2] If the lowercase of the string = "yes",
      go add 'passage-of-time did obscure evidence' to answers
      and go add 'passage-of-time did cause prejudice' to reports
      and return,
    otherwise if the lowercase of the string = "no",
      go add 'passage-of-time did not obscure evidence' to answers
      and return,
    otherwise if the string = "-",
      go add 'if-passage-of-time did obscure evidence' to unknowns
      and go add 'passage-of-time did not obscure evidence' to assumptions.

End.


To determine-prejudice-from-late-notice:

[1] Send {2 lines,"[",the qnum,"]",2 blanks,
      "Is the government able to show that proper notice",cr,
      "would have resulted in cheaper resolution of the condition",cr,
      "by CO? (Yes/No)",cr,": "}
      and assert dpfln is a question
      and go input-yes-no
      and increment 'the qnum'.

[2] If the lowercase of the string = "yes",
      go add 'improper-notice did cause additional cost' to answers
      and go add 'additional-cost did cause prejudice' to reports
      and return,
    otherwise if the lowercase of the string = "no",
      go add 'improper-notice did not cause additional cost' to answers
      and return,
    otherwise if the string = "-",
      go add 'if-improper-notice did cause additional cost' to unknowns
      and go add 'improper-notice did not cause additional cost'
                                to assumptions.

End.


To determine-other-prejudice:

[1] Send {2 lines,"[",the qnum,"]",2 blanks,
      "Can the government show that it suffered any other prejudice",cr,
```

```
      "through inadequacy of or lack of notice? (Yes/No)",cr,": "}
      and assert dop is a question
      and go input-yes-no
      and increment 'the qnum'.

[2] If the lowercase of the string = "yes",
      go add 'something-else did cause prejudice' to answers
      and go add 'something-else did cause prejudice' to reports
      and return,
    otherwise if the lowercase of the string = "no",
      go add 'nothing-else did cause prejudice' to answers
      and return,
    otherwise if the string = "-",
      go add 'if-anything-else did cause prejudice' to unknowns
      and go add 'nothing-else did cause prejudice' to assumptions.

End.


To determine-difficulty-of-defending-claim:

[1] Send {2 lines,"[".the qnum,"]",2 blanks,
      "Was the government defense made impossible by inadequacy",cr,
      "of or lack of notice? (Yes/No)",cr,": "}
     and assert ddodc is a question
     and go input-yes-no
     and increment 'the qnum'.

[2] If the lowercase of the string = "yes",
      go add 'defense-against claim was made impossible' to answers
      and send {2 lines,"Inadequacy of or lack of notice plus",cr,
            "resulting prejudice to government render",cr,
            "entitlement to DSC unlikely.",cr,
            "However, I will continue our analysis.",cr}
      and return,
    otherwise if the lowercase of the string = "no",
      go add 'defense-against claim was not made impossible'
                                                to answers
      and return,
    otherwise if the string = "-",
      go add 'if-defense-against claim was made impossible'
                                                to unknowns
      and go add 'defense-against claim was not made impossible'
                                                to assumptions.

End.


To decide-proof-of-claim:

[1] Send {2 lines,"[",the qnum,"]",2 blanks,
      "Was the contractor able to provide sufficient additional",cr,
      "proof of claim sufficient to prove entitlement to",cr,
```

```
        "DSC claim in spite of slight prejudice to the",cr,
        "government? (Yes/No)",cr,": "}
        and assert dpoc is a question
        and go input-yes-no
        and increment 'the qnum'.

[2] If the lowercase of the string = "yes",
        go add 'additional-proof does exist to-prove-entitlement' to answers
        and send {2 lines,"The contractor may be entitled to the claim in",cr,
              "spite of the resulting prejudice suffered by ",cr,
              "the government.",cr}
        and return,
    otherwise if the lowercase of the string = "no",
        go add 'additional-proof does not exist to-prove-entitlement'
                                                        to answers
        and send {2 lines,"Inadequacy of or lack of notice plus resulting",cr,
              "prejudice to the government render entitlement",cr,
              "to DSC claim unlikely.  However, I will continue",cr,
              "our analysis.",cr}
        and return,
    otherwise if the string = "-",
        go add 'if-additional-proof does exist to-prove-entitlement'
                                                        to unknowns
        and go add 'additional-proof does exist to-prove-entitlement'
                                                        to assumptions.

End.
```

```
[: QUESTS2 parsed Tue Nov 22 19:44:28 1983 by Kruppen :]

[***************************** OBLIGATION ********************************]

To determine-obligation:

[1] Send {3 lines,"[",the qnum,"]",2 blanks,
        "Who expressly assumed obligation for or risk of the",cr,
              "condition in the contract?",cr,
      "a.  The government   ",cr,
      "b.  The contractor   ",cr,
      "c.  Both the government and the contractor ",cr,
      "d.  Neither the government nor the contractor ",cr,":  "}
    and assert do-1 is a question
    and let the nchoices be 4
    and go read-vrbl
    and increment 'the qnum'.

[2] If the vrbl = "a",
      go add 'government did expressly assume obligation-risk' to answers
      and send {3 lines,"The government expressly assumed the obligation",cr,
                "or risk of this condition, therefore, relief may come",cr,
                "from the clause assigning this obligation or risk.",cr,
                "Further analysis is beyond my capabilities.",cr}
      and go add 'contract did assign obligation-risk' to reports
      and return,
    otherwise if the vrbl = "b",
      go add 'contractor did expressly assume obligation-risk' to answers
      and send {1 line,"The contractor has assumed the obligation or",cr,
                "risk of encountering the condition.",cr}
      and deny entitlement is still-probable
      and assert entitlement is not still-probable
      and return,
    otherwise if the vrbl = "c",
      go add 'confusion-about-who did assume risk' to answers
      and send {1 line,"There is confusion as to who assumed responsibility",
            cr,"for the condition, however, to continue our analysis",cr,
            "I will assume that neither the government nor the",cr,
            "contractor assumed the risk.",cr}
      and go add 'government did not expressly assume obligation-risk'
                                   to assumptions
      and go add 'contractor did not expressly assume obligation-risk'
                                   to assumptions
      and return,
    otherwise if the vrbl = "d",
      go add 'government did not expressly assume obligation-risk'
                                   to answers
      and go add 'contractor did not expressly assume obligation-risk'
                                       to answers
      and go add 'contract did not assign obligation-risk' to reports
      and return,
    otherwise if the vrbl = "-",
```

```
    go add 'government did not expressly assume obligation-risk'
                                                      to assumptions
    and go add 'contractor did not expressly assume obligation-risk'
                                                      to assumptions
    and go add 'which-party did assume risk' to unknowns.

End.


[*************************** EXCLUDED CONDITIONS *****************************]

To determine-physical-conditions:

[1] Send {3 lines,"[",the qnum,"]",2 blanks,
         "Is the condition in question directly related to the",cr,
       "physical conditions at the work site? (Yes/No)",cr,": "}
    and assert dpc is a question
    and go input-yes-no
    and increment 'the qnum'.

[2] If the lowercase of the string = "yes",
    go add 'condition is directly-related-to-physical-conditions'
                                                      to answers
    and return,
  otherwise if the lowercase of the string = "no",
    go add 'condition is not directly-related-to-physical-conditions'
                                                      to answers
    and send {3 lines,"Differing site conditions claim is limited to ",cr,
                    "conditions and causes that are directly related",cr,
                    "to physical conditions at the site.",cr,
               "Condition is probably beyond scope of clause.",cr}
    and deny entitlement is still-probable
    and assert entitlement is not still-probable
    and return,
  otherwise if the string = "-",
    go add 'if-condition is directly-related-to-physical-conditions'
                                                      to unknowns
    and go add 'condition is directly-related-to-physical-conditions'
                                                      to assumptions.

End.


To determine-static-physical-condition:

[1] Send {2 lines,"[",the qnum,"]",2 blanks,
         "Is the condition a static physical part of the",cr,
                "work site? (Yes/No)",cr, ": "}
    and assert dspc is a question
    and go input-yes-no
    and increment 'the qnum'.

[2] If the lowercase of the string = "yes",
```

```
    go add 'condition is static physical part of-work-site' to answers
    and return,
  otherwise if the lowercase of the string = "no",
    go add 'condition is not static physical part of-work-site' to answers
    and send {3 lines,"Differing site condition is limited to conditions",
               cr,"which are a static physical part of the work.",cr,
               "site. Recovery may be possible through another",cr,
               "clause or through breach.",cr}
    and deny entitlement is still-probable
    and assert entitlement is not still-probable
    and return,
  otherwise if the string = "-",
    go add 'if-condition is static physical part of-work-site' to unknowns
    and go add 'condition is static physical part of-work-site'
                                      to assumptions.

End.


To determine-condition-occurrence:

[1] Send {2 lines,"[",the qnum,"]",2 blanks,
    "Did the condition occur before or after the contract",cr,
              "award?",cr}
    and assert dco is a question
    and go input-before-after
    and increment 'the qnum'.

[2] If the lowercase of the string = "before",
    go add 'condition did occur before contract award' to answers
    and return,
  otherwise if the lowercase of the string = "after",
    go add 'condition did occur after contract award' to answers
    and send {1 line,"Differing site condition is usually limited to",cr,
          "those conditions occurring before bid or contract award.",
              1 line,"However, an exception may be allowed.",cr}
    and return,
  otherwise if the lowercase of the string = "-",
    go add 'if-condition did occur before contract award' to unknowns
    and go add ' condition did occur before contract award'
                                      to assumptions.

End.


[*************************** AFTER-BID CONDITIONS ***************************]

To determine-cause:

[1] Send {3 lines,"[",the qnum,"]",2 blanks,
    "What or who did the differing site condition result",cr,
         "from? (If you are unfamiliar with this question type '?'",cr,
```

```
                "then hit return)",cr,": "}
        and assert dc is a question
        and increment 'the qnum'.

[2] Read {anything (bind the answer),cr}
      and until the lowercase of the answer = "end.",
        assert dc is a question
          and if the answer = "?",
                go explain-quests,
            otherwise if the answer = "-",
                go add 'what-condition did result-from' to unknowns
                and go add 'condition did result from no-one'
                                        to assumptions
                and go count-unknowns
                and forget about the question
                and return,
            otherwise if the answer = "q",
                assert system is stopped
                and forget about the question
              and go wrap-up,
            otherwise if the answer = "??",
                go explain-logic,
            otherwise if the answer is valid-answer,
                evaluate {"go add 'condition did result from ",
                        the answer,"' to answers."}
                and forget about the question,
            otherwise,
                send {"Improper form...please type '?' for insructions.",
                        cr},
                and send {1 line,": "}
                and read {anything (bind the answer),cr}.

End.


To determine-differing-quantities:

[1] Send {2 lines,"[",the qnum,"]",2 blanks,
      "Is the condition one of differing quantities? (Yes/No)",
                                    cr,": "}
      and assert dfq is a question
      and go input-yes-no
      and increment 'the qnum'.

[2] If the lowercase of the string = "yes",
      go add 'condition is differing quantity' to answers
      and return,
    otherwise if the lowercase of the string = "no",
      go add 'condition is not differing quantity' to answers
      and return,
    otherwise if the string = "-",
      go add 'if-condition is differing quantity' to unknowns
      and go add 'condition is not differing quantity' to assumptions.
```

```
End.


To determine-clause-for-differing-quantities:

[1] Send {2 lines,"[",the qnum,"]",2 blanks,
      "Does the contract have a variation in estimated",cr,
           "quantity clause? (Yes/No)",cr,": "}
      and assert dcfdq is a question
      and go input-yes-no
      and increment 'the qnum'.

[2] If the lowercase of the string = "yes",
      go add 'contract does have var-in-est-quant-clause' to answers
      and send {3 lines,"Since there is a variation in estimated quantity",
           cr,"clause in the contract and the condition is one of",cr,
           "differing quantities it is best to seek relief",cr,
           "through the variation in estimated quantity clause.",1 line,
           "At this time I am unable to assist your analysis",cr,
           "any further."}
      and deny entitlement is still-probable
      and assert entitlement is not still-probable
      and return,
    otherwise if the lowercase of the string = "no",
      go add 'contract does not have var-in-est-quant-clause'
                                              to answers
      and return,
    otherwise if the string = "-",
      go add 'if-contract does have var-in-est-quant-clause'
                                         to unknowns
      and go add 'contract does not have var-in-est-quant-clause'
                                         to assumptions.

End.


To determine-interaction-of-condition:

[1] Send {2 lines,"[",the qnum,"]",2 blanks,
      "Was the condition caused by an interaction of a",cr,
           "non-compensable condition with physical factors",cr,
      "at the site? (Yes/No)",cr,": "}
      and assert dioc is a question
      and go input-yes-no
      and increment 'the qnum'.

[2] If the lowercase of the string = "yes",
      go add 'condition did result from
            interacting-non-compensable-and-physical-factors'
                                              to answers
      and return,
    otherwise if the lowercase of the string = "no",
      go add 'condition did not result from
```

```
                interacting-non-compensable-and-physical-factors'
                                                  to answers
        and return,
      otherwise if the string = "-",
        go add 'if-condition did result from
                interacting-non-compensable-and-physical-factors'
                                                  to unknowns
        and go add 'condition did not result from
               interacting-non-compensable-and-physical-factors'
                                                  to assumptions.

End.


To determine-government-control:

[1] Send {2 lines,"[",the qnum,"]",2 blanks,
        "Did the government have control over the actions",cr,
              "of the third party which is at fault? (Yes/No)",cr,": "}
        and assert dgc is a question
        and go input-yes-no
        and increment 'the qnum'.

[2] If the lowercase of the string = "yes",
        go add 'government did have control over third party' to answers
        and return,
      otherwise if the lowercase of the string = "no",
        go add 'government did not have control over third party'
                                                  to answers
        and return,
      otherwise if the string = "-",
        go add ' if-government did have control over third party'
                                              to unknowns
        and go add 'government did not have control over third party'
                                                  to assumptions.'

End.


To determine-adequate-control:

[1] Send {2 lines,"[",the qnum,"]",2 blanks,
           "Did the government exercise adequate control over",cr,
        "the actions of the third party? (Yes/No)",cr,": "}
        and assert dac is a question
        and go input-yes-no
        and increment 'the qnum'.

[2] If the lowercase of the string = "yes",
        go add 'government did exercise adequate control' to answers,
      otherwise if the lowercase of the string = "no",
        go add 'government did not exercise adequate control' to answers,
      otherwise if the string = "-",
```

go add 'if-government did exercise adequate control' to unknowns
and go add 'government did exercise adequate control' to assumptions.

End.

```
[: QUESTS3 parsed Tue Dec  6 03:58:02 1983 by Kruppen :]

[********************* EXPRESS/IMPLIED DEFICIENCIES *********************]

To determine-express-conditions:

[1] Send {3 lines,"[",the qnum,"]",2 blanks,
       "Were there affirmatively expressed statements in the",cr,
      "contract concerning the relevant subsurface or ",cr,
      "latent conditions at the site? (Yes/No)",cr,": "}
      and assert dec is a question
      and go input-yes-no
      and increment 'the qnum'.

[2] If the lowercase of the string = "yes",
       go add 'contract did contain statements concerning condition'
                                          to answers
       and go add 'contract did contain indications concerning condition'
                                              to reports
       and return,
     otherwise if the lowercase of the string = "no",
       go add 'contract did not contain statements concerning condition'
                                              to answers
       and return,
     otherwise if the string = "-",
       go add 'if-contract did contain statements concerning condition'
                                              to unknowns
       and go add 'contract did not contain statements concerning condition'
                                              to assumptions.

End.


To determine-implied-conditions:

[1] Send {2 lines,"[",the qnum,"]",2 blanks,
         "Would the affirmatively expressed contract statements",cr,
       "on the general conditions at the site lead a reasonable ",cr,
       "contractor to believe that the condition could be",cr,
       "expected? (Yes/No)",cr,": "}
       and assert dic-1 is a question
       and go input-yes-no
       and increment 'the qnum'.

[2] If the lowercase of the string = "yes",
       go add 'contract did contain general conditions implying condition'
                                              to answers
       and go add 'contract did contain indications concerning condition'
                                              to reports
       and return,
     otherwise if the lowercase of the string = "no",
       go add 'contract did not contain general
```

```
                         conditions implying condition'
                                        to answers
     and return,
   otherwise if the string = "-",
     go add 'if-contract did contain general
                         conditions implying condition'
                                        to unknowns
     and go add 'contract did not contain general
                         conditions implying condition'
                                        to assumptions.

End.


To determine-inferred-conditions:

[1] Send {2 lines,"[",the qnum,"]",2 blanks,
     "Were indications of subsurface or latent conditions",cr,
     "inferred from the contract as a whole? (Yes/No)",cr,": "}
     and assert dic-2 is a question
     and go input-yes-no
     and increment 'the qnum'.

[2] If the lowercase of the string = "yes",
       go add 'contract did lead one-to infer conditions' to answers
       and return,
     otherwise if the lowercase of the string = "no",
       go add 'contract did not lead one-to infer conditions' to answers
       and return,
     otherwise if the string = "-",
       go add 'if-contract did lead one-to infer conditions' to unknowns
       and go add 'contract did not lead one-to infer conditions'
                                        to assumptions.

End.


To determine-if-inference-justified:

[1] Send {2 lines,"[",the qnum,"]",2 blanks,
         "Was this inference justified based on the information",cr,
       "in the contract? (Yes/No)",cr,": "}
      and assert diij is a question
      and go input-yes-no
      and increment 'the qnum'.

[2] If the lowercase of the string = "yes",
      go add 'contract information did justify inference' to answers
      and go add 'contract did contain indications concerning condition'
                                        to reports
      and return,
    otherwise if the lowercase of the string = "no",
      go add 'contract information did not justify inference' to answers
```

```
    and return,
  otherwise if the string = "-",
    go add 'if-contract information did justify inference' to unknowns
    and go add 'contract information did not justify inference'
                                                   to assumptions.

End.
```

[*************** LATENT DEFICIENCIES/CONTRACT INDICATIONS ******************]

```
To determine-latent-deficiencies:

[1] Send {3 lines,"[",the qnum,"]",2 blanks,
         "Were there latent deficiencies in the contract",cr,
           "indications? (Yes/No)",cr,": "}
    and assert dld is a question
    and go input-yes-no
    and increment 'the qnum'.

[2] If the lowercase of the string = "yes",
    go add 'contract indications did contain latent deficiencies'
                                              to answers
    and return,
  otherwise if the lowercase of the string = "no",
    go add 'contract indications did not contain latent deficiencies'
                                              to answers
    and deny entitlement is still-probable
    and assert entitlement is not still-probable
    and return,
  otherwise if the string = "-",
    go add 'if-contract indications did contain latent deficiencies'
                                              to unknowns
    and go add 'contract indications did contain latent deficiencies'
                                              to assumptions.

End.

To determine-physical-data:

[1] Send {2 lines,"[",the qnum,"]",2 blanks,
         "Was the physical data made a part of the",cr,
           "contract? (Yes/No)",cr,": "}
    and assert dpd is a question
    and go input-yes-no
    and increment 'the qnum'.

[2] If the lowercase of the string = "yes",
    go add 'contract information did contain physical data'
                                          to answers
    and return,
  otherwise if the lowercase of the string = "no",
```

```
        go add 'contract information did not contain physical data'
                                                  to answers
        and return,
      otherwise if the string = "-",
        go add 'if-contract information did contain physical data'
                                                  to unknowns
        and go add 'contract information did not contain physical data'
                                                  to assumptions.

End.


To determine-value-of-other-data:

[1] If latent-deficiencies was checked,
        send {2 lines,"[",the qnum,"]",2 blanks,
                "Would this physical data have revealed or resolved",cr,
              "the apparent deficiencies in the contract indications for",cr,
              "a reasonable contractor? (Yes/No)",cr,": "},
      otherwise if contract-indications was checked,
        send {2 lines,"[",the qnum,"]",2 blanks,
               "Would this physical data have indicated the nature of ",cr,
            "existing conditions to a reasonable contractor? (Yes/No)",
              cr,": "},
      and assert dvood is a question
      and go input-yes-no
        and increment 'the qnum'.

[2] If the lowercase of the string = "yes",
        go add 'contract information did reveal deficiencies/conditions'
                                                       to answers
        and deny entitlement is still-probable
        and assert entitlement is not still-probable
        and return,
      otherwise if the lowercase of the string = "no",
        go add 'contract information did not reveal deficiencies/conditions'
                                                       to answers
        and go add 'contractor is not required-to inspect other data'
                                                        to reports
        and return,
      otherwise if the string = "-",
        go add 'if-contract information did reveal deficiencies/conditions'
                                                       to unknowns
        and go add 'contract information did
                    not reveal deficiencies/conditions'
                                                       to assumptions.

End.

[********************** STANDARD OF COMPARISON ************************]

To determine-establishment-of-standard:
```

```
[1] Send {3 lines,"[",the qnum,"]",2 blanks,
      "By what means can the contractor establish expected conditions?",cr,
         "a.  Customs of the trade",cr,
         "b.  General knowledge in the industry",cr,
         "c.  Manufacturers instructions or recommendations",cr,
         "d.  None of the above ",cr,":  "}
      and assert deos is a question
      and let the nchoices be 4
      and go read-vrbl
      and increment 'the qnum'.

[2] If the vrbl = "a",
      go add 'expected conditions are establishable from customs-of-trade'
                                          to answers,
    otherwise if the vrbl = "b",
      go add 'expected conditions are establishable
                        from general-knowledge-in-industry'
                                          to answers,
    otherwise if the vrbl = "c",
      go add 'expected conditions are establishable
                        from manufacturers-recommendations'
                                          to answers,
    otherwise if the vrbl = "d",
      go add 'expected conditions are not establishable' to answers
      and deny entitlement is still-probable
      and assert entitlement is not still-probable
      and return,
    otherwise if the vrbl = "-",
      go add 'if-expected conditions are establishable' to unknowns
      and go add 'contractor is able-to
                        establish standard-of-expectation'
                                          to assumptions.

End.

To determine-different-from-standard:

[1] Send {2 lines,"[",the qnum,"]",2 blanks,
      "Does the actual condition differ from the standard?  (Yes/No)",
         cr,":  "}
      and assert ddfs is a question
      and go input-yes-no
      and increment 'the qnum'.

[2] If the lowercase of the string = "yes",
      go add 'condition does differ from standard-of-expectation'
                                          to answers
      and return,
    otherwise if the lowercase of the string = "no",
      go add 'condition does not differ from standard-of-expectation'
                                              to answers
      and deny entitlement is still-probable
```

```
    and assert entitlement is not still-probable
    and return,
  otherwise if the string = "-",
    go add 'if-condition does differ from standard-of-expectation'
                                          to unknowns
    and go add 'condition does differ from standard-of-expectation'
                                         to assumptions.

End.


[*************************** SUPERIOR KNOWLEDGE ***************************]

To determine-knowledge:

[1] If government is a party-in-question,
      send {3 lines,"[",the qnum,"]",2 blanks,
      "Did the ",the party-in-question," know about the",cr,
      the knowledge,"? (Yes/No)",cr,": "},
  otherwise if contractor is a party-in-question,
    (if contract-deficiency is a knowledge,
         send {3 lines,"[",the qnum,"]",2 blanks,
          "Was the contractor in possession of unilateral superior",cr,
          "or imputed knowledge that the actual conditions patently",cr,
          "differed from the indications in the contract? (Yes/No)",cr,
           ": "},
       otherwise send {3 lines,"[",the qnum,"]",2 blanks,
          "Was the contractor in possession of unilateral superior",cr,
          "knowledge of the actual conditions? (Yes/No)",cr,": "}),
  and assert dk is a question
  and go input-yes-no
  and increment 'the qnum'.

[2] If the lowercase of the string = "yes",
    evaluate {"Go add '",the party-in-question," was aware-of ",
                              the knowledge,"' to answers."}
    and forget about the party-in-question
    and return,
  otherwise if the lowercase of the string = "no",
    evaluate {"Go add '",the party-in-question," was not aware-of ",
                              the knowledge,"' to answers."}
    and forget about the party-in-question
    and return,
  otherwise if the string = "-",
    evaluate {"Go add 'if-",the party-in-question," was aware-of ",
                              the knowledge,"' to unknowns."}
    and evaluate {"Go add '",the party-in-question," was not aware-of ",
                        the knowledge,"' to assumptions."}
    and forget about the party-in-question.

End.
```

```
To determine-communication-of-knowledge:

[1] Send {2 lines,"[",the qnum,"]",2 blanks,
      "Did the government reveal ",the knowledge," to the contractor?",
           cr,"(Yes/No)",cr,":  "}
      and assert dcok is a question
      and go input-yes-no
      and increment 'the qnum'.

[2] If the lowercase of the string = "yes",
      evaluate {"Go add 'government did reveal ",the knowledge,
                                        "' to answers."}
      and return,
    otherwise if the lowercase of the string = "no",
      evaluate {"Go add 'government did not reveal ",the knowledge,
                                        "' to answers."}
      and go add 'government did have superior knowledge' to reports
      and send {3 lines,"Oversight, misrepresentation or concealment on",cr,
                "the part of the government.  A possible",cr,
                "breach has occurred.  However, I will",cr,
                "continue with the analysis."}
      and return,
    otherwise if the string = "-",
      evaluate {"Go add 'if-government did reveal ",the knowledge,
                                        "' to unknowns."}
      and evaluate {"Go add 'government did reveal ",the knowledge,
                                        "' to assumptions."}.

End.


To determine-evidence:

[1] Send {2 lines,"[",the qnum,"]",2 blanks,
      "Does evidence exist to prove that ",the statement,"?  (Yes/No)",
                cr,":  "}
      and assert de is a question
      and go input-yes-no
      and increment 'the qnum'.

[2] If the lowercase of the string = "yes",
      evaluate {"Go add 'proof does indicate ",the statement,"' to answers."}
      and return,
    otherwise if the lowercase of the string = "no",
      evaluate {"Go add 'proof does not indicate ",the statement,
                                        "' to answers."}
      and forget about the statement
      and return,
    otherwise if the string = "-",
      evaluate {"Go add 'if-proof does indicate ",the statement,
                                        "' to unknowns."}
      and evaluate {"Go add 'proof does not indicate ",the statement,
                                        "' to assumptions."}
```

```
and forget about the statement
and return.

[3] If the lowercase of the string = "yes",
    if contractor-had-knowledge is a statement,
        go add 'contractor did have superior knowledge' to reports,
    otherwise if government-communicated-knowledge is a statement,
        go add 'contractor did have imputed knowledge-of-condition'
                                        to reports,
  and forget about statement.

End.


To determine-effect-of-simple-inquiry:

[1] Send {2 lines,"[",the qnum,"]",2 blanks,
        "Would simple inquiries by the contractor have ",cr,
      "revealed the condition to be contrary to the ",cr,
          the bit-o-info,"? (Yes/No)",cr,": "}
    and assert deosi is a question
    and go input-yes-no
    and increment 'the qnum'.

[2] If the lowercase of the string = "yes",
    go add 'simple inquiry did give potential to-know contrary conditions'
                                                to answers
    and send {3 lines,"It has become obvious that the contractor did",cr,
             "not make simple inquiries.",cr}
    and deny entitlement is still-probable
    and assert entitlement is not still-probable
    and return,
  otherwise if the lowercase of the string = "no",
    go add 'simple inquiry did not give potential
                  to-know contrary conditions'
                          to answers
    and return,
  otherwise if the string = "-",
    go add 'if-simple inquiry did give potential
                  to-know contrary conditions'
                          to unknowns
    and go add 'simple inquiry did not give potential
                  to-know contrary conditions'
                          to assumptions.

End.


[***************************** SITE INSPECTION ****************************]

To determine-hinderance-of-inspection:

[1] Send {3 lines,"[",the qnum,"]",2 blanks,
```

```
        "Did any of the following acts of the government hinder",cr,
           "the site inspection?",2 lines,
           "a.  Access to site was denied ",cr,
           "b.  Inadequate time was allowed ",cr,
           "c.  No act of the government hindered inspection ",cr,":  "}
    and assert dhti is a question
    and let the nchoices be 3
    and go read-vrbl
    and increment 'the qnum'.

[2] If the vrbl = "a",
       go add 'government did withhold access-to-site' to answers
       and go add 'site-inspection is not required' to reports
       and return,
    otherwise if the vrbl = "b",
       go add 'government did allow inadequate time' to answers
       and go add 'site-inspection is not required' to reports
       and return,
    otherwise if the vrbl = "c",
       go add 'government did not hinder inspection' to answers
       and return,
    otherwise if the vrbl = "-",
       go add 'if-government did hinder inspection' to unknowns
       and go add 'government did not hinder inspection' to assumptions.

End.


To determine-if-inspection-made:

[1] Send {2 lines,"[",the qnum,"]",2 blanks,
       "Did the contractor conduct a site inspection? (Yes/No)",
                                    cr,":  "}
       and assert diim is a question
       and go input-yes-no
       and increment 'the qnum'.

[2] If the lowercase of the string = "yes",
       go add 'contractor did conduct site-inspection' to answers
       and return,
    otherwise if the lowercase of the string = "no",
       go add 'contractor did not conduct site-inspection' to answers
       and go add 'contractor did not conduct site-inspection' to reports
       and send {2 lines,"Your contractor risked encountering an unexpected",cr,
             "condition that a reasonable inspection might",cr,
             "have revealed.",cr}
       and return,
    otherwise if the string = "-",
       go add 'if-contractor did conduct site-inspection' to unknowns
       and go add 'contractor did conduct site-inspection' to assumptions.

End.
```

```
To determine-reasonable-inspection:

[1] Send {3 lines,"[",the qnum,"]",2 blanks,
      "Was the inspection conducted by the contractor a ",cr,
      "reasonable inspection, i.e., one comparable to that",cr,
      "expected from a reasonable, prudent contractor",cr,
      "experienced in that particular field of work? (Yes/No)",
      cr,": "}
      and assert dri is a question
      and go input-yes-no
      and increment 'the qnum'.

[2] If the lowercase of the string = "yes",
      go add 'site-inspection was reasonable inspection' to answers
      and go add 'reasonable inspection did not reveal condition' to reports
      and return,
    otherwise if the lowercase of the string = "no",
      go add 'site-inspection was not reasonable inspection' to answers
      and return,
    otherwise if the string = "-",
      go add 'if-site-inspection was reasonable inspection' to unknowns
      and go add 'site-inspection was reasonable inspection'
                                        to assumptions
      and go add 'reasonable inspection did not reveal condition'
                                        to reports.

End.


To determine-unreasonable-inspection:

[1] Send {3 lines,"[",the qnum,"]",2 blanks,
      "Would the actual condition have been discernable ",cr,
      "by a layman contractor performing a reasonable ",cr,
      "inspection? (Yes/No)",cr,": "}
      and assert dui is a question
      and go input-yes-no
      and increment 'the qnum'.

[2] If the lowercase of the string = "yes",
      go add 'reasonable inspection did have potential-to-reveal condition'
                                           to answers
      and return,
    otherwise if the lowercase of the string = "no",
      go add 'reasonable inspection did not have potential-to-reveal condition'
                                           to answers
      and go add 'reasonable inspection is not required-for-entitlement'
                                           to reports
      and return,
    otherwise if the string = "-",
      go add 'if-reasonable inspection did have
                     potential-to-reveal condition'
                                           to unknowns
```

```
        and go add 'reasonable inspection did not
                         have potential-to-reveal condition'
                                               to assumptions
        and go add 'reasonable inspection is not required-for-entitlement'
                                               to reports.

End.

[********************************** RELIANCE **********************************]

To determine-reliance:

[1] Send {3 lines,"[",the qnum,"]",2 blanks,
           "Did the contractor reasonably rely on ",the statement,
        "when preparing his bid causing him to suffer prejudice",cr,
        "through this reliance? (Yes/No)",cr,": "}
        and assert dr is a question
        and go input-yes-no
        and increment 'the qnum'.

[2] If the lowercase of the string = "yes",
        go add 'contractor did suffer prejudice through reliance' to answers
        and return,
      otherwise if the lowercase of the string = "no",
        go add 'contractor did not suffer prejudice through reliance'
                                            to answers
        and return,
      otherwise if the string = "-",
        go add 'if-contractor did suffer prejudice through reliance'
                                            to  unknowns
        and go add 'contractor did suffer prejudice through reliance'
                                               to assumptions.

End.

[************************** MATERIAL DIFFERENCE **************************]

To determine-material-difference:

[1] Send {3 lines,"[",the qnum,"]",2 blanks,
           "Was the difference between actual and ",the statement,cr,
           "conditions a material difference? (Yes/No)",cr,": "}
        and assert dmd is a question
        and go input-yes-no
        and increment 'the qnum'.

[2] If the lowercase of the string = "yes",
        go add 'difference was material' to answers
        and return,
      otherwise if the lowercase of the string = "no",
        go add 'difference was not material' to answers
```

```
    and deny entitlement is still-probable
    and assert entitlement is not still-probable
    and send {2 lines,"The contractor should have anticipated the ",cr,
             "condition as ",the comment,".",cr}
    and return,
  otherwise if the string = "-",
    go add 'if-difference was material' to unknowns
    and go add 'difference was material' to assumptions.

End.

[*********************** EXCULPATORY LANGUAGE ***************************]

To determine-exculpatory-clause-exists:

[1] If exculpatory-language-I was checked,
    send {3 lines,"[",the qnum,"]",2 blanks,
          "Is there an exculpatory clause denying any government",cr,
      "liability and responsibility for actual conditions",cr,
      "different from those indicated in the contract",cr,
      "documents? (Yes/No)",cr,": "},
  otherwise if exculpatory-language-II was checked,
    send {2 lines,"[",the qnum,"]",2 blanks,
             "Is there a clause in the contract placing the risk",cr,
         "of conditions not revealed by the contractor's site",cr,
         "inspection? (Yes/No)",cr,": "},
    and assert dece is a question
    and go input-yes-no
    and increment 'the qnum'.

[2] If the lowercase of the string = "yes",
    go add 'contract does contain exculpatory-clause' to answers
    and return,
  otherwise if the lowercase of the string = "no",
    go add 'contract does not contain exculpatory-clause' to answers
    and send {3 lines,"Since the contract does not contain an exculpatory",
             cr,"clause entitlement will probably be allowed.",
             2 lines,"Consult your local CO for further details.",
             cr}
    and return,
  otherwise if the string = "-",
    go add 'if-contract does contain exculpatory-clause' to unknowns
    and go add 'contract does not contain exculpatory-clause'
                                    to assumptions.

End.

To determine-specific-to-clause:

[1] Send {2 lines,"[",the qnum,"]",2 blanks,
    "Is the exculpatory clause specific to the DSC clause and to",cr,
```

```
            "the condition encountered? (Yes/No)",cr,": "}
        and assert dstc is a question
        and go input-yes-no
        and increment 'the qnum'.

[2] If the lowercase of the string = "yes",
        go add 'exculpatory-clause is specific-to DSC clause' to answers
        and return,
    otherwise if the lowercase of the string = "no",
        go add 'exculpatory-clause is not specific-to DSC clause' to answers
        and go add 'validity-of exculpatory-clause is in_doubt' to reports
        and return,
    otherwise if the string = "-",
        go add 'if-exculpatory-clause is specific-to DSC clause' to unknowns
        and go add 'exculpatory-clause is specific-to DSC clause'
                                                    to assumptions.

End.


To determine-clarity-of-clause:


[1] Send {2 lines,"[",the qnum,"]",2 blanks,
        "Are the language and intent of the exculpatory clause",cr,
                "clear and unambiguous? (Yes/No)",cr,": "}
        and assert dcoc is a question
        and go input-yes-no
        and increment 'the qnum'.

[2] If the lowercase of the string = "yes",
        go add 'exculpatory-clause is clear-and-unambiguous' to answers
        and return,
    otherwise if the lowercase of the string = "no",
        go add 'exculpatory-clause is not clear-and-unambiguous' to answers
        and return,
    otherwise if the string = "-",
        go add 'if-exculpatory-clause is clear-and-unambiguous' to unknowns
        and go add 'exculpatory-clause is clear-and-unambiguous'
                                                to assumptions.

End.
```

```
[: STARTUP parsed Tue Dec  6 04:27:00 1983 by Kruppen :]


To get-started:

Execute Cyclically.

[1] Unless files are loaded,
        reset ROSIELOUDFLG
        and load UTILS
        and set ROSIELOUDFLG
        and activate status
        and restore status
        and deactivate.

[2] Unless dbases are created,
        go create-dbases.

[3] Unless files are loaded,
        send {1 lines,"I am loading the necessary files for the analysis",cr,
                "it may take some time so make yourself comfortable.",
                2 lines}
        and reset ROSIELOUDFLG
        and load each of quests, unk-quests and drivers
        and load each of quests2, unk-quests2 and drivers2
        and load each of quests3, unk-quests3 and drivers3
        and load each of entitle, dscasdata and exp-quests
        and load each of conclude1, conclude2, conclude3,
                conclude4 and conclude5
        and set ROSIELOUDFLG
        and go explain-system.

[4] If system is restarting,
        send {2 lines,
          "******************* SESSION RESTARTING *******************",
          2 lines,"Since you interupted me I'll have to backtrack",cr,
                "a bit.  Let's see as I remember...."}
        and if complete-analysis was run,
          go locate-place.

[5] Unless system is restarting,
        go clean-up
        and reset ROSIELOUDFLG
        and load dscasdata
        and set ROSIELOUDFLG
        and send {2 lines,control "G G G","I'm ready to get started...",
                                                1 line}.

[6] Unless system is restarting,
        activate status
        and if there is a previous-session,
          send {1 line,"Do you wish to continue a previous ",cr,
                "session? (Yes/No)", cr,": "}
```

```
        and deactivate
        and assert gs-1 is a question
        and go read-yes-no
        and (if the lowercase of the string = "yes",
              go load-session,
         otherwise if the lowercase of the string = "no"
              go name-session),
      otherwise,
        deactivate
        and go name-session.

[7] Unless system is restarting,
      send {1 line,"Would you like me to record our discussion so that you",cr,
        "may obtain a printout of it to view at a later date? (Yes/No)",
              cr,": "}
      and assert gs-2 is a question
      and go read-yes-no
      and if the lowercase of the string = "yes",
        assert record-of-session is being-kept
        and if there is a current-session,
              evaluate {"dribble to ",the current-session,"."}
              and send {"The name of this session shall be: ",
                              the current-session},
         otherwise,
              evaluate {"dribble to ",the session-name,"."}
              and send {"The name of this session shall be: ",
                              the session-name}.

[8] Unless system is restarting,
      if there is a current-session,
        activate status
        and send {3 lines,"Author of previous session: "}
        and evaluate {"display the author-of-",the current-session,
                                " and send {cr}."}
        and send {"Date of previous session: "}
        and evaluate {"display the date-of-",the current-session,
                                " and send {cr}."}
        and evaluate {"Deny ",the rply," is a current-session."}
        and deactivate.

[9] Unless system is restarting,
      send {3 lines,
              "You may perform any one of the following analyses:",2 lines,
        "DSC Assertion          Final Payment          Notice Form",cr,
        "Notice Promptness      Responsible Receiver   Government Prejudiced",2 lines,
        "Contract Obligation    Excluded Conditions    After-bid Conditions",2 lines,
        "   Express-implied Conditions     Contract Indications",cr,
        "   Latent Deficiencies            Standard Conditions",cr,
        "   Site Inspection",2 lines,
        "   Superior Knowledge-I           Superior Knowledge-II",cr,
        "   Reliance I                     Reliance II",cr,
        "   Material Difference-I          Material Difference-II",cr,
        "   Exculpatory Langauge-I         Exculpatory Language-II",2 lines,
```

```
" Complete Analysis            No Module",2 lines,
"I will begin the analysis as soon as you have selected any ONE",cr,
"of the options listed. Please enter the selected option",cr,
"EXACTLY as it appears in the list followed by a '.'",cr,": "}
and go get-option.

[10] Unless no-module is a module,
        unless system is restarting,
           send {1 line,20 blanks,"SESSION IN PROGRESS",3 lines},
     and evaluate {"Go check-",the module,"."}.

[11] Unless complete-analysis was run,
     unless no-module is a module,
        evaluate {"Go draw-",the module,"-conclusions."}
        and go check-entitlement.

[12] Go wrap-up.

End.


To wrap-up:

[1] If analysis is stopped,
      send {2 lines,
        "******************** SESSION INTERRUPTED ********************",
        cr,25 blanks,"(by user)",cr},
    otherwise,
      send {2 lines,
        "********************* END OF ANALYSIS *********************",
        cr,8 blanks,"(You should receive my bill in 7-10 days.)",cr,
        14 blanks,"(PROMPT PAYMENT IS APPRECIATED)",cr}.

[2] Send {3 lines,"Do you wish to see the results from this session? (Yes/No)",
                                    cr,": "}
      and assert ss is a question
      and go read-yes-no
      and if the lowercase of the string = "yes",
        send {1 line,"Which of the following results do you wish",cr,
                     "to see?",cr}
      and go examine-results.

[3] If analysis is stopped,
    send {1 line,"Do you wish to continue our analysis? (Yes/No)",2 blanks,
          "Note: If you choose not to I will quit this session.",
               cr,": "}
    and assert ss is a question
    and go read-yes-no
    and deny analysis is stopped
    and if the lowercase of the string = "yes",
          assert system is restarting
          and go get-started,
      otherwise if the lowercase of the string = "no",
```

```
        assert system is quitting.

[4] Send {1 line,"Do you wish to store this session? (Yes/No)",cr,": "}
    and assert ss is a question
    and go read-yes-no
    and if the lowercase of the string = "yes",
        go store-cycle
        and send {1 line,"Session has been saved as requested.",
                                                    2 lines},
     otherwise,
        send {1 line,"Session NOT saved.",2 lines}.

[5] If record-of-session is being-kept,
    if there is a session-name,
       evaluate {"stop dribbling."}
       and send {2 lines,"Oh yes...You can get a printout of our",cr,
            "session by typing 'print' followed by the session",cr,
            "name.ORIG once you have logged out of ROSIE.",cr}
       and evaluate {"Copy ",the session-name," to ",
                 the session-name,".ORIG and delete ",
                 the session-name,"."},
    otherwise,
       evaluate {"stop dribbling."}
       and send {2 lines,"Oh yes...You can get a printout of our",cr,
            "session by typing 'print' followed by the session",cr,
            "name once you have logged out of ROSIE.",cr}.

[6] If the ucount = 12,
    send {2 lines,"Nice chatting with you.",2 lines,
         "Get your facts straight next time....",cr}
    and quit.

[7] If system is quitting,
        send {2 lines,"If you wish to run another analysis please",cr,
             "type 'go get-started.'.",cr}
        and quit.

[8] Send {2 lines,"Do you wish to run another analysis? (Yes/No)",cr,": "}
    and assert gs-3 is a question
    and go read-yes-no.

[9] If the lowercase of the string = "yes",
    send {1 line,"Good... But you'll have to give me a minute or",cr,
              "so to tidy things up a bit.",cr}
    and (if system is restarting,
           deny system is restarting,
         otherwise do nothing),
   otherwise if the lowercase of the string = "no",
    send {1 line,"It has been a pleasure chatting with you, we'll",cr,
              "have to do it again sometime.",cr}
    and quit.

End.
```

To explain-system:

```
[1] Send {2 lines,
        "   Hello, you are about to begin interacting with the",cr,
        "first version of the Differing Site Condition Analysis ",cr,
        "System (DSCAS). My purpose is to analyze the differing",cr,
        "site condition claim. I will lead you through the analysis",cr,
        "process as painlessly as possible and attempt to give you",cr,
        "some assistance in deciding whether or not your contractor",cr,
        "has a good chance for entitlement through the DSC claim.",cr}.

[2] Send {2 lines,"Do you wish to see the instructions? (Yes/No)",cr,": "}
    and assert ss is a question
    and go read-yes-no
    and if the lowercase of the string = "no",
     send {1 line,"Alright.",cr,
          "I'll be ready to start in a couple of seconds...",cr}
     and return.

[3] Send {2 lines,"The instructions are as follows:",2 lines,
      "   I will ask you questions and you answer them, simple",cr,
      "right. The order in which I ask you questions is dependent",cr,
      "on the answers which you give and my understanding of a ",cr,
      "lawyer's analysis process. Each of the questions includes",cr,
      "either a list of choices to select or the yes/no prompt ",cr,
      "(Yes/No). Other valid answers for all questions are: ",2 lines,
      " 'q' -- The quit option allowing you to stop ",cr,
      "       momentarily to check the results of the session",cr,
      "       thus far or to stop completely.",2 lines,
      " '-' -- The unknown option to be used when you do not know ",cr,
      "       the answer to the question being asked. I must ",cr,
      "       caution you that after 11 unknowns I get extremely",cr,
      "       frustrated with your incompetence and will stop the",cr,
      "       analysis at the next unknown.",2 lines}.

[4] Send {"  Other answers which are valid are:",cr,
      " '?' -- The 'help' option allows you to receive some advice",cr,
      "       on the meaning of various choices or the format a",cr,
      "       particular answer must be in. NOTE -- This option ",cr,
      "       is not available for ANY of the questions which ",cr,
      "       require a Yes/No answer.",cr,
      " '??' - The 'inquire' option allows you to inquire about",cr,
      "       the logic which I am following, i.e., Why I am",cr,
      "       asking that particular question. NOTE -- THIS ",cr,
      "       OPTION IS NOT YET AVAILABLE ON CURRENT VERSION.",
                          2 lines}.

[5] Send {"  One other important point which I must mention is that",cr,
      "due to a bug in the code on which I am built you must use",cr,
      "ONLY the DELETE key for correcting errors on input. Press",cr,
      "the delete key gently one press at a time and try not to ",cr,
      "use it more than a few times per session. If by chance you",cr,
      "end up getting thrown out of our analysis process you will",cr,
```

```
    "see a ':' (ignore the message with it). To restart our ",cr,
    "analysis simply press a SHIFT '6' (the up arrowhead '^'),",cr,
    "this will allow me to continue.",2 lines,
    "Now if you will excuse me I'll get back to the preparation",cr,
    "for our session.",cr}.

End.


To restart:

[1] Assert system is restarting
          and activate
          and go get-started.

End.


[rule 1] Wait for 5 seconds and send {2 lines,"Hello Diek.",2 lines,
         "Would you like to run an analysis?",cr,": "}
         and read {anything (bind the rp),cr}
         and if the lowercase of the rp = "yes",
           forget about the rp
                and send {2 lines,"I'll get things started for you.",cr}
                         and go get-started,
          otherwise,
            forget about the rp
            and send {2 lines,"That's too bad, I enjoyed our last chat.",
                         cr,"So long...",cr}.
```

```
[: UNK-QUESTS parsed Thu Dec  1 21:37:07 1983 by Kruppen :]

[****************************************************************************]

[************************** RULESETS TO CHECK UNKNOWNS **********************]

To unknown-DSC-assertion:

[1] Send {2 lines,"Assertion of DSC was unknown at last session.",cr,
      "Do you have any more information regarding",cr,
      "this point? (Yes/No)",cr,":  "}
            and assert didwa is a question
            and go read-yes-no.

[2] If the lowercase of the string = "yes",
            go remove 'if-DSC was asserted' from unknowns
            and go determine-if-DSC-was-asserted,
        otherwise if the lowercase of the string = "no"
            go add 'DSC was asserted' to assumptions.

End.

[*********************************************]

To unknown-final-payment:

[1] Send {2 lines,"Occurence of final-payment was unknown at last session.",
            cr,"Do you have any more information regarding this",cr,
            "point? (Yes/No)",cr,":  "}
     and assert difpwm is a question
     and go read-yes-no.

[2] If the lowercase of the string = "yes",
         go remove 'if-final-payment was made' from unknowns
         and go determine-if-final-payment-was-made,
     otherwise if the lowercase of the string = "no",
         go add 'final-payment was not made' to assumptions.

End.

[********************************************]

To unknown-form-of-notice:

[1] Send {2 lines,"The form of notice was unknown when I last spoke",cr,
      "with you.  Do you know anything more about",cr,
      "this yet? (Yes/No)",cr,":  "}
            and assert dfon is a question
            and go read-yes-no.
```

```
[2] If the lowercase of the string = "yes",
            go remove 'form-of-notice is unknown' from unknowns
            and go determine-form-of-notice,
        otherwise if the lowercase of the string = "no",
            go add 'form-of-notice was other' to assumptions
          and go add 'contractor did constructively
                      comply with notice-requirement-**'
                                to reports.

End.


To unknown-other-notice:

[1] Send {2 lines,"Last time we spoke you were not sure whether or not",cr,
      "other circumstances led to waiver of the notice",cr,
      "requirement.  Do you have an answer to this yet?  (Yes/No)",cr,
      ":  "}
      and assert don is a question
      and go read-yes-no.

[2] If the lowercase of the string = "yes",
      go remove 'if-notice requirement was waived' from unknowns
      and go determine-other-notice
      and return,
    otherwise if the lowercase of the string = "no",
      go add 'notice-requirement was not waived' to assumptions
      and go add 'actual-occurrence-of-notice is in_doubt' to reports.

End.

{**********************************************}

To unknown-potential-promptness:

[1] Send {2 lines,"I beleive the last time I spoke with you you had ",cr,
      "not determined whether the contractor should have",cr,
            "have been aware of the differing site condition at",cr,
      "an earlier time. ",cr,
      "Have you reached a decision regarding this point?  (Yes/No)",
                cr,":  "}
      and assert dp-2 is a question
      and go read-yes-no.

[2] If the lowercase of the string = "yes",
      go remove 'if-contractor did have potential-to-know condition sooner'
                                  from unknowns
      and send {1 line,"Good...",1 line}
      and go determine-potential-promptness
      and return,
    otherwise if the lowercase of the string = "no",
      go add 'contractor did not have potential-to-know condition sooner'
```

```
                                        to assumptions
        and go add 'notice was prompt-**' to reports.

End.


To unknown-promptness:

[1] Send {2 lines,"Last time we spoke you were unable to pinpoint",cr,
            "the exact time that notice was given relative ",cr,
            "to the progression of work. ",cr,
            "Have you determined when notice did occur? (Yes/No)",cr,
               ": "}
      and assert dp-1 is a question
      and go read-yes-no.

[2] If the lowercase of the string = "yes",
        go remove 'if-notice was given before condition disturbed'
                                        from unknowns
        and go determine-promptness
        and return,
      otherwise if the lowercase of the string = "no",
        send {"You should really check into some of this stuff soon.."}
        and go add 'notice was given before condition disturbed'
                                        to assumptions
        and go add 'notice was given before remedial work performed'
                                        to assumptions
        and go add 'notice was prompt-**' to reports.

End.

[*******************************************]

To unknown-receiver:

[1] Send {2 lines,"Do you know to whom notice of the DSC was",cr,
            "given yet? (Yes/No)",cr,": "}
      and assert drr is a question
      and go read-yes-no.

[2] If the lowercase of the string = "yes",
      go remove 'to-whom was notice given' from unknowns
      and go determine-responsible-receiver
      and return,
    otherwise if the lowercase of the string = "no",
      go add 'receipt-of-notice is in_doubt' to reports
      and go add 'notice was not received by government' to assumptions.

End.
```

```
To unknown-position:

[1] Send {2 lines,"Do you know yet whether or not the other government",cr,
          "employee, to whom notice was given, was cognisant ",cr,
          "of the implications of the condition? (Yes/No)",cr,
          ": "}
    and assert dp is a question
    and go read-yes-no.

[2] If the lowercase of the string = "yes",
    go remove 'if-government employee did understand implications'
                                   from unknowns
    and go determine-position
    and return,
  otherwise if the lowercase of the string = "no",
    go add 'government employee did understand implications'
                                   to assumptions.

End.


To unknown-communication-of-notice:

[1] Send {2 lines,"Are you still uncertain as to whether or not the",cr,
          "government employee who received notice of the",cr,
          "DSC told his CO? (Yes/No)",cr,": "}
    and assert doe-1 is a question
    and go read-yes-no.

[2] If the lowercase of the string = "yes",
    go remove 'if-government employee did communicate DSC awareness to-CO'
                                        from unknowns
    and go determine-other-employee
    and return,
  otherwise if the lowercase of the string = "no",
    go add 'government employee did communicate DSC awareness to-CO'
                                        to assumptions.

End.


To unknown-proof-of-DSC-notice:

[1] Send {2 lines,"Have you found any evidence yet which proves that",cr,
          "notice of the DSC was given to some other ",cr,
          "government employee? (Yes/No)",cr,": "}
    and assert doe-2 is a question
    and go read-yes-no.

[2] If the lowercase of the string = "yes",
    go remove 'if-proof-of DSC notice does exist' from unknowns
    and assert proof is being-checked
    and go determine-other-employee
```

```
        and forget about being-checked
        and return,
      otherwise if the lowercase of the string = "no",
        go add 'proof-of DSC notice does exist' to assumptions
        and go add 'government was aware of DSC-**' to reports.

End.


[*************************************************]

To unknown-prejudice-from-passage-of-time:

[1] Send {2 lines,"During our last session you were not sure if any",cr,
        "evidence against the claim had been obscurred due to the",cr,
        "passage of time.",2 lines,
        "Do you know enough to answer this question yet? (Yes/No)",
             cr,": "}
        and assert dfpot is a question
        and go read-yes-no.

[2] If the lowercase of the string = "yes",
        go remove 'if-passage-of-time did obscure evidence' from unknowns
        and go determine-prejudice-from-passage-of-time
        and return,
      otherwise if the lowercase of the string = "no",
        go add 'passage-of-time did not obscure evidence' to assumptions.

End.


To unknown-prejudice-from-late-notice:

[1] Send {2 lines,"Do you know yet whether or not proper notice",cr,
             "would have resulted in cheaper resolution of",cr,
             "the claim by the CO? (Yes/No)",cr,": "}
              and assert dpfln is a question
              and go read-yes-no.

[2] If the lowercase of the string = "yes",
              go remove 'if-improper-notice did cause additional cost'
                                          from unknowns
              and go determine-prejudice-from-late-notice
              and return,
            otherwise if the lowercase of the string = "no",
              go add 'improper-notice did not cause additional cost'
                                          to assumptions.

End.


To unknown-other-prejudice:
```

```
[1] Send {2 lines,"Last time we met you were unsure if the government",cr,
        "could show that it had suffered any other prejudice",cr,
        "through improper notice.",cr,
        "Do you have any more information regarding this",cr,
        "question yet? (Yes/No)",cr,": "}
        and assert dop is a question
        and go read-yes-no.

[2] If the lowercase of the string = "yes",
        go remove 'if-anything-else did cause prejudice' from unknowns
        and go determine-other-prejudice
        and return,
    otherwise if the lowercase of the string = "no",
        send {2 lines,"Don't you think you had better look into it??",2 lines}
        and go add 'nothing-else did cause prejudice' to assumptions.

End.


To unknown-difficulty-of-defending-claim:

[1] Send {2 lines,"Have you received any more information as to whether",cr,
        "or not improper notice did increase the difficulty",cr,
        "of defending against the claim? (Yes/No)",cr,": "}
        and assert ddodc is a question
        and go read-yes-no.

[2] If the lowercase of the string = "yes",
        go remove 'if-defense-against claim was made impossible'
                                        from unknowns
        and go determine-difficulty-of-defending-claim
        and return,
    otherwise if the lowercase of the string = "no",
        go add 'defense-against claim was not made impossible'
                                        to assumptions.

End.


To unknown-proof-of-claim:

[1] Send {2 lines,"At our last meeting you were not sure if the contractor",cr,
        "could provide sufficient additional proof of the claim",cr,
        "to prove entitlement to the claim.",cr,
        "Do you have an answer to this question yet? (Yes/No)",cr,": "}
        and assert dpoc is a question
        and go read-yes-no.

[2] If the lowercase of the string = "yes",
        go remove 'if-additional-proof does exist to-prove-entitlement'
                                        from unknowns
        and go determine-proof-of-claim
        and return,
```

otherwise if the lowercase of the string = "no",
go add 'additional-proof does exist to-prove-entitlement'
to assumptions.

End.

```
[: UNK-QUESTS2 parsed Tue Nov 22 19:49:48 1983 by Kruppen :]

[**************************************************]

To unknown-obligation:

[1] Send {2 lines,"Do you know yet who expressly assumed obligation",cr,
      "for or risk of the condition in the contract? (Yes/No)",
      cr,": "}
      and assert do-1 is a question
      and go read-yes-no.

[2] If the lowercase of the string = "yes",
      if 'confusion-about-who did assume risk' is true in answers,
            go remove 'confusion-about-who did assume risk' from answers
            and go determine-obligation
            and return,
       otherwise,
            go remove 'which-party did assume risk' from unknowns
            and go determine-obligation.

[3] If the lowercase of the string = "no",
      go add 'government did not expressly assume obligation-risk'
                                          to assumptions
      and go add'contractor did not expressly assume obligation-risk'
                                          to assumptions
      and if 'confusion-about-who did assume risk' is true in answers,
            send {2 lines,"There is still confusion as to who assumed ",cr,
          "responsibility for the condition, however, to ",cr,
          "continue our analysis process I will assume that",cr,
          "neither the government nor the contractor assumed",cr,
          "responsibility.",2 lines},
      and return.

End.

[**************************************************]

To unknown-physical-conditions:

[1] Send {2 lines,"Have you found out yet whether or not the condition",cr,
      "is directly related to the physical conditions at",cr,
      "the work site? (Yes/No)",cr,": "}
      and assert dpc is a question
      and go read-yes-no.

[2] If the lowercase of the string = "yes",
      go remove 'if-condition is directly-related-to-physical-conditions'
                                         from unknowns
      and go determine-physical-conditions
      and return,
```

```
    otherwise if the lowercase of the string = "no",
      go add 'condition is directly-related-to-physical-conditions'
                                    to assumptions
      and send {2 lines,"Don't you think you should begin looking ",cr,
           "into a few of these unknowns...",2 lines}.

End.


To unknown-static-physical-condition:

[1] Send {2 lines,"When we chatted last you were not sure if the",cr,
      "condition was actually a static physical part",cr,
      "of the work site.",cr,
      "Do you know yet? (Yes/No)",cr,": "}
      and assert dspc is a question
      and go read-yes-no.

[2] If the lowercase of the string = "yes",
      go remove 'if-condition is static physical part of-work-site'
                                    from unknowns
      and go determine-static-physical-condition
      and return,
    otherwise if the lowercase of the string = "no",
      go add 'condition is static physical part of-work-site'
                                    to assumptions.

End.


To unknown-condition-occurrence:

[1] Send {2 lines,"Do you know yet whether the condition occurred",cr,
      "before or after the contract award? (Yes/No)",cr,": "}
      and assert dco is a question
      and go read-yes-no.

[2] If the lowercase of the string = "yes",
      go remove 'if-condition did occur before contract award'
                                    from unknowns
      and go determine-condition-occurrence
      and return,
    otherwise if the lowercase of the string = "no",
      go add 'condition did occur before contract award'
                                    to assumptions.

End.

[**************************************************]

To unknown-cause:
```

```
[1] Send {2 lines,"You were unsure as to what or who the condition",cr,
        "resulted from.",cr,
          "Do you know yet? (Yes/No)",cr,": "}
      and assert dc is a question
      and go read-yes-no.

[2] If the lowercase of the string = "yes",
      go remove 'what-condition did result-from' from unknowns
      and go determine-cause
      and return,
    otherwise if the lowercase of the string = "no",
      go add 'condition did result from no-one' to assumptions.

End.


To unknown-differing-quantities:

[1] Send {2 lines,"Do you know yet whether or not the condition is one",cr,
      "of differing quantities? (Yes/No)",cr,": "}
      and assert ddq is a question
      and go read-yes-no.

[2] If the lowercase of the string = "yes",
      go remove 'if-condition is differing quantity' from unknowns
      and go determine-differing-quantities
      and return,
    otherwise if the lowercase of the string = "no",
      go add 'condition is not differing quantity' to assumptions.

End.


To unknown-clause-for-differing-quantities:

[1] Send {2 lines,"Have you read your contract yet to determine",cr,
      "whether or not it contains a variation in",cr,
      "estimated quantity clause? (Yes/No)",cr,": "}
      and assert dcfdq is a question
      and go read-yes-no.

[2] If the lowercase of the string = "yes",
      go remove 'if-contract does have var-in-est-quant-clause'
                                        from unknowns
      and go determine-clause-for-differing-quantities
      and return,
    otherwise if the lowercase of the string = "no",
           go add 'contract does not have var-in-est-quant-clause'
                                        to assumptions.

End.
```

```
To unknown-interaction-of-condition:

[1] Send {2 lines,"Have you found out yet whether of not the condition",cr,
      "was caused by an interaction of a non-compensable ",cr,
      "condition with a physical condition at the site? (Yes/No)",
           cr,": "}
      and assert dioc is a question
      and go read-yes-no.

[2] If the lowercase of the string = "yes",
      go remove 'if-condition did result from
           interacting-non-compensable-and-physical-factors'
                                        from unknowns
      and go determine-interaction-of-condition
      and return,
   otherwise if the lowercase of the string = "no",
      go add 'condition did not result from
           interacting-non-compensable-and-physical-factors'
                                        to assumptions.

End.


To unknown-government-control:

[1] Send {2 lines,"Do you know yet whether or not the government did ",cr,
              "exercise control over the actions of the ",cr,
              "third party? (Yes/No)",cr,": "}
         and assert dgc is a question
         and go read-yes-no.

[2] If the lowercase of the string = "yes",
         go remove 'if-government did have control over third party'
                                    from unknowns
         and go determine-government-control
         and return,
   otherwise if the lowercase of the string = "no",
         go add 'government did not have control over third party'
                                    to assumptions.

End.


To unknown-adequate-control:

[1] Send {2 lines,"When we last spoke you were not sure whether or not",cr,
      "the government had adequately exercised the control",cr,
      "it had over the actions of the third party. ",cr,
      "Do you have an answer to this yet? (Yes/No)",cr,": "}
      and assert dac is a question
      and go read-yes-no.

[2] If the lowercase of the string = "yes",
```

```
    go remove 'if-government did exercise adequate control'
                                              from unknowns
    and go determine-adequate-control
    and return,
  otherwise if the lowercase of the string = "no",
    go add 'government did exercise adequate control'
                                              to assumptions.

End.
```

```
[: UNK-QUESTS3 parsed Thu Dec  1 21:51:10 1983 by Kruppen :]

[*******************************************]

To unknown-express-conditions:

[1] Send {2 lines,"When we last spoke you were unsure as to whether ",cr,
      "the contract contained affirmatively expressed ",cr,
      "statements concerning the relevant site subsurface",cr,
      "or latent conditions.",cr,
      "Have you checked for this in the contract yet? (Yes/No)",
          cr,": "}
      and assert dec is a question
      and go read-yes-no.

[2] If the lowercase of the string = "yes",
      go remove 'if-contract did contain statements concerning condition'
                                         from unknowns
      and go determine-express-conditions
      and return,
    otherwise if the lowercase of the string = "no",
      go add 'contract did not contain statements concerning condition'
                                         to asssumptions.


End.


To unknown-implied-conditions:

[1] Send {2 lines,"Do you know yet whether or not the affirmatively ",cr,
      "expressed contract statements on the general",cr,
      "conditions would lead a reasonable contractor ",cr,
      "to believe that the condition could be expected? (Yes/No)",
                cr,": "}
      and assert dic-1 is a question
      and go read-yes-no.

[2] If the lowercase of the string = "yes",
      go remove 'if-contract did contain general
                          conditions implying condition'
                                         from unknowns
      and go determine-implied-conditions
      and return,
    otherwise if the lowercase of the string = "no",
      go add 'contract did not contain general
                         conditions implying condition'
                                         to asumptions.

End.
```

```
To unknown-inferred-conditions:

[1] Send {2 lines,"Last time we met you said you were uncertain",cr,
     "as to whether or not the indications of subsurface",cr,
     "or latent conditions were inferred from the contract ",cr,
     "as a whole.",cr,
     "Do you have an answer for this question yet? (Yes/No)",
          cr,": "}
     and assert dic-2 is a question
     and go read-yes-no.

[2] If the lowercase of the string = "yes",
     go remove 'if-contract did lead one-to infer conditions'
                                   from unknowns
     and go determine-inferred-conditions
     and return,
  otherwise if the lowercase of the string = "no",
     go add 'contract did not lead one-to infer conditions'
                                   to assumptions.

End.


To unknown-if-inference-justified:

[1] Send {2 lines,"Do you know yet whether or not the inference ",cr,
     "concerning the condition was justified based",cr,
     "on the information in the contract? (Yes/No)",cr,": "}
     and assert diij is a question
     and go read-yes-no.

[2] If the lowercase of the string = "yes",
     go remove 'if-contract information did justify inference'
                                   from unknowns
     and go determine-if-inference-justified
     and return,
  otherwise if the lowercase of the string = "no",
     go add 'contract information did not justify inference'
                                   to assumptions.

End.


[*********************************************]

To unknown-latent-deficiencies:

[1] Send {2 lines,"Do you know yet whether or not there were latent",cr,
     "deficiencies in the contract indications? (Yes/No)",
               cr,": "}
     and assert dld is a question
     and go read-yes-no.
```

```
[2] If the lowercase of the string = "yes",
      go remove 'if-contract indications did contain
                                    latent deficiencies'
                                                  from unknowns
      and go determine-latent-deficiencies
      and return,
   otherwise if the lowercase of the string = "no",
      go add 'contract indications did contain latent deficiencies'
                                                  to assumptions.

End.


To unknown-physical-data:

[1] Send {2 lines,"Do you know yet whether or not the physical data ",cr,
      "was made a part of the contract? (Yes/No)",cr,": "}
      and assert dpd is a question
      and go read-yes-no.

[2] If the lowercase of the string = "yes",
      go remove 'if-contract information did contain physical data'
                                                  from unknowns
      and go determine-physical-data
      and return,
   otherwise if the lowercase of the string = "no",
      go add 'contract information did not contain physical data'
                                                  to assumptions.

End.


To unknown-value-of-other-data:

[1] If latent-deficiencies was checked,
      send {2 lines,"Do you know yet whether or not the additional",cr,
      "data in the contract would have revealed or resolved",cr,
      "deficiencies in the contract indications? (Yes/No)",
            cr,": "},
   otherwise if the contract-indications was checked,
      send {2 lines,"Do you know yet whether or not the additional ",cr,
      "data in the contract would have indicated the nature",cr,
      "of the existing conditions? (Yes/No)",cr,": "},
      and assert dvood is a question
      and go read-yes-no.

[2] If the lowercase of the string = "yes",
      go remove 'if-contract information did reveal
                                    deficiencies/conditions'
                                                  from unknowns
      and go determine-value-of-other-data
      and return,
   otherwise if the lowercase of the string = "no",
```

```
        go add 'contract information did not reveal
                                deficiencies/conditions'
                                        to assumptions.

End.

[**********************************************]

To unknown-establishment-of-standard:

[1] Send {2 lines,"When we last spoke you were unsure as to the means ",cr,
        "by which the contractor could establish expected ",cr,
        "conditions. ",cr,
        "Have you discovered an answer to this question yet? (Yes/No)",
            cr,": "}
        and assert deos is a question
        and go read-yes-no.

[2] If the lowercase of the string = "yes",
        go remove 'if-expected conditions are establishable'
                                    from unknowns
        and go determine-establishment-of-standard
        and return,
      otherwise if the lowercase of the string = "no",
        go add 'contractor is able-to establish standard-of-expectation'
                                        to assumptions.

End.

To unknown-different-from-standard:

[1] Send {2 lines,"Do you know yet whether or not the actual condition ",cr,
        "differs from the standard? (Yes/No)",cr,": "}
        and assert ddfs is a question
        and go read-yes-no.

[2] If the lowercase of the string = "yes",
        go remove 'if-condition does differ from standard-of-expectation'
                                    from unknowns
        and go determine-different-from-standard
        and return,
      otherwise if the lowercase of the string = "no",
        go add 'condition does differ from standard-of-expectation'
                                    to assumptions.

End.

[**********************************************]

To unknown-knowledge:
```

```
[1] Send {2 lines,"Do you know yet whether or not the ",
          the party-in-question," knew",cr,
      "about or anticipated the ",the knowledge,"? (Yes/No)",
          cr,": "}
      and assert dk is a question
      and go read-yes-no.

[2] If the lowercase of the string = "yes",
      evaluate {"go remove 'if-",the party-in-question," was aware-of ",
                                 the knowledge,"' from unknowns."}
      and go determine-knowledge
      and return,
    otherwise if the lowercase of the string = "no",
      evaluate {"go add '",the party-in-question," was not aware-of ",
                             the knowledge,"' to assumptions."}
      and forget about party-in-question.

End.


To unknown-communication-of-knowledge:

[1] Send {2 lines,"When we last spoke you were not sure if the ",cr,
          "government did reveal the ",the knowledge,
          "to the contractor. ",cr,
          "Do you have an answer to this yet? (Yes/No)",cr,": "}
           and assert dcok is a question
           and go read-yes-no.

[2] If the lowercase of the string = "yes",
           evaluate {"go remove 'if-government did reveal ",the knowledge,
                                       "' from unknowns."}
           and go determine-communication-of-knowledge
           and return,
         otherwise if the lowercase of the string = "no",
           evaluate {"go add 'government did reveal ",the knowledge,
                                       "' to assumptions."}.

End.


To unknown-evidence:

[1] Send {2 lines,"Last time we spoke you were not sure if evidence ",cr,
      "existed to prove that the ",the statement,".",cr,
      "Do you know the answer to this yet? (Yes/No)",cr,": "}
      and assert de is a question
      and go read-yes-no.

[2] If the lowercase of the string = "yes",
      evaluate {"go remove 'if-proof does indicate ",the statement,
                                "' from unknowns."}
      and go determine-evidence
```

```
      and return,
    otherwise if the lowercase of the string = "no",
      evaluate {"go add 'proof does not indicate ",the statement,
                                       "' to assumptions."}
      and forget about the statement.

End.


To unknown-effect-of-simple-inquiry:

[1] Send {2 lines,"Do you know yet whether or not simple inquiries by ",cr,
           "the contractor would have revealed the condition",cr,
           "to be contrary to the ",the bit-o-info,"? (Yes/No)",cr,": "}
       and assert deosi is a question
       and go read-yes-no.

[2] If the lowercase of the string = "yes",
      go remove 'if-simple inquiry did give potential
                     to-know contrary conditions'
                                    from unknowns
      and go determine-effect-of-simple-inquiry
      and return,
    otherwise if the lowercase of the string = "no",
      go add 'simple inquiry did not give potential
                     to-know contrary conditions'
                                    to assumptions.

End.


[*********************************************]

To unknown-hinderance-of-inspection:

[1] Send {2 lines,"Do you know yet whether or not any acts of the ",cr,
           "government may have hindered the site inspection? (Yes/No)",
                 cr,": "}
           and assert dhoi is a question
           and go read-yes-no.

[2] If the lowercase of the string = "yes",
           go remove 'if-government did hinder inspection' from unknowns
           and go determine-hinderance-of-inspection
           and return,
         otherwise if the lowercase of the string = "no",
           go add 'government did not hinder inspection' to assumptions.

End.


To unknown-if-inspection-made:
```

```
[1] Send {2 lines,"Do you know yet whether or not the contractor ",cr,
      "conducted a site investigation? (Yes/No)",cr,": "}
      and assert diim is a question
      and go read-yes-no.

[2] If the lowercase of the string = "yes",
      go remove 'if-contractor did conduct site inspection'
                                              from unknowns
      and go determine-if-site-inspection-made
      and return,
    otherwise if the lowercase of the string = "no",
      go add 'contractor did conduct site investigation'
                                              to assumptions.

End.

[**********************************************]

To unknown-reasonable-inspection:

[1] Send {2 lines,"Do you know yet whether or not the contractor ",cr,
      "conducted a reasonable site inspection? (Yes/No)",
          cr,": "}
      and assert dri is a question
      and go read-yes-no.

[2] If the lowercase of the string = "yes",
      send {2 lines,"Good......",2 lines}
      and go remove 'if-site-inspection was reasonable inspection'
                                                       from unknowns
      and go determine-reasonable-inspection
      and return,
    otherwise if the lowercase of the string = "no",
      go add 'site-inspection was reasonable inspection'
                                              to assumptions.

End.

To unknown-unreasonable-inspection:

[1] Send {2 lines,"Do you know yet whether or not a reasonable site ",cr,
      "inspection would have revealed the condition? (Yes/No)",
                cr,": "}
      and assert dri is a question
      and go read-yes-no.

[2] If the lowercase of the string = "yes",
      go remove 'if-reasonable inspection did have
                           potential-to-reveal condition'
                                                       from unknowns
      and go determine-unreasonable-inspection
```

```
        and return,
      otherwise if the lowercase of the string = "no",
        go add 'reasonable inspection did not have
                      potential-to-reveal condition'
                                  to assumptions
        and go add 'reasonable inspection is not
                      required-for-entitlement-**'
                                  to reports.

   End.
```

[**]

To unknown-reliance:

```
[1] Send {2 lines,"Last time we spoke you were unsure if the contractor ",cr,
        "had suffered prejudice by relying on the ",
        the statement," when preparing his bid. ",cr,
        "Do you have an answer to this yet? (Yes/No)",cr,": "}
        and assert dr is a question
        and go read-yes-no.

[2] If the lowercase of the string = "yes",
        go remove 'if-contractor did suffer prejudice through reliance'
                                  from unknowns
        and go determine-reliance
        and return,
      otherwise if the lowercase of the string = "no",
        go add 'contractor did suffer prejudice through reliance'
                                  to assumptions.

End.
```

[***]

To unknown-material-difference:

```
[1] Send {2 lines,"Last time we spoke you did not know whether or ",cr,
        "not the actual conditions were materially different ",cr,
        "from the ",the statement,"conditions. ",2 lines,
        "Do you know the answer to this yet? (Yes/No)",cr,": "}
        and assert dmd is a question
        and go read-yes-no.

[2] If the lowercase of the string = "yes",
        go remove 'if-difference was material' from unknowns
        and go determine-material-difference
        and return,
      otherwise if the lowercase of the string = "no",
        go add 'difference was material' to assumptions.
```

```
End.

[*************************************************]

To unknown-exculpatory-clause-exists:

[1] Send {2 lines,"Do you know yet whether or not the contract ",cr,
      "contains an exculpatory clause? (Yes/No)",cr,": "}
      and assert dece is a question
      and go read-yes-no.

[2] If the lowercase of the string = "yes",
      go remove 'if-contract does contain exculpatory-clause'
                                   from unknowns
      and go determine-exculpatory-clause-exists
      and return,
   otherwise if the lowercase of the string = "no",
      go add 'contract does not contain exculpatory-clause'
                                   to assumptions
      and send {2 lines,"Let's get with it... All you need to do",cr,
                "is read the contract.",2 lines}.

End.

To unknown-specific-to-clause:

[1] Send {2 lines,"Do you know yet if the exculpatory clause is ",cr,
      "specific to the DSC clause? (Yes/No)",cr,": "}
      and assert dstc is a question
      and go read-yes-no.

[2] If the lowercase of the string = "yes",
      go remove 'if-exculpatory-clause is specific-to DSC clause'
                                          from unknowns
      and go determine-specific-to-clause
      and return,
   otherwise if the lowercase of the string = "no",
      go add 'exculpatory-clause is specific-to DSC clause'
                                          to assumptions.

End.

To unknown-clarity-of-clause:

[1] Send {2 lines,"Last time we spoke you were not sure if the ",cr,
      "language and intent of the clause were clear and",cr,
      "unambiguous. ",2 lines,
      "Do you have an answer to this yet? (Yes/No)",cr,": "}
      and assert dcoc is a question
      and go read-yes-no.
```

```
[2] If the lowercase of the string = "yes",
      go remove 'if-exculpatory-clause is clear-and-unambiguous'
                                                    from unknowns
      and go determine-clarity-of-clause
      and return,
    otherwise if the lowercase of the string = "no",
      go add 'exculpatory-clause is clear-and-unambiguous'
                                                    to assumptions.

End.
```

```
[: UTILS parsed Tue Dec  6 03:55:00 1983 by Kruppen :]

[*****************************************************************************]
[************************ Rulesets to read input ************************]

To get-date:

Execute cyclically.

[1] Send {1 line,"[",the qnum,"]", 2 blanks,
                    "On what date did the ",the event,
                    " occur?  (month day, year)",
                                cr,":  "}
            and assert gd-1 is a question
            and read { {anything (bind the entry),cr}|
                                cr (bind the entry)}.

[2] If the entry = "-",
            evaluate {"go add '<unknown> is a date-of-",
                                the event,"' to answers.",cr}
            and forget about the question
               and forget about the event
            and go count-unknowns
            and return,
      otherwise if the entry = "?",
            go explain-date,
      otherwise if the entry = "q",
            assert analysis is stopped
            and forget about the question
            and go wrap-up,
      otherwise if the entry is valid,
            evaluate {"add '<",the entry,"> is a date-of-",
                                      the event,"' to answers."}
            and forget about the event
            and forget about the question
            and return,
      otherwise,
            send {2 lines,"Improper date form, please try again.",
                                                  2 lines}.

End.


To input-yes-no:

Execute cyclically.

[1] Read {{0 or more blanks,1 or more nonblanks (bind the string),cr}|
                          cr(bind the string)}
          and if (the lowercase of the string = "yes"
            or the lowercase of the string = "no"),
            forget about the question
            and return,
```

```
        otherwise if the string = "-",
          go count-unknowns
          and forget about the question
          and return,
        otherwise if the string = "?",
          send {1 line,
                "Sorry no explanation for questions with yes/no answers.",
                                    1 line,": "},
        otherwise if the string = "??",
             send {1 line,
                "I thought I told you that option is not ready yet...",
                     1 line,": "},
        otherwise if the string = "q",
          assert analysis is stopped
          and forget about the question
          and go wrap-up,
        otherwise,
          send {2 lines,"Just answer 'Yes','No','??' or '-' PLEASE!.",
                               2 lines,": "}.

End.


To read-yes-no:

Execute cyclically.

[1] Read {{0 or more blanks,1 or more nonblanks (bind the string),cr}|
              cr (bind the string)}
       and if (the lowercase of the string = "yes"
                 or the lowercase of the string = "no"),
          forget about the question
          and return,
        otherwise if the string = "?",
          send {1 line,
               "Sorry no explanation for questions with yes/no answers.",
                          1 line},
        otherwise if the string = "??",
             send {1 line,
                "I thought I told you that option is not ready yet...",
                     1 line,": "},
        otherwise if the string = "q",
          assert analysis is stopped
          and forget about the question
          and go wrap-up,
        otherwise,
          send {2 lines,"Just answer 'Yes','No' or '??' PLEASE!.",
                          2 lines,": "}.

End.


To read-vrbl:
```

Execute cyclically.

```
[1] Read {{0 or more blanks, 1 or more nonblanks (bind the vrbl),cr}|
                        cr (bind the vrbl)}
        and if (the vrbl = "a"
                or the vrbl = "b"
                 or (the vrbl = "c"
                     and the nchoices ~ < 3)
                  or (the vrbl = "d"
                      and the nchoices ~ < 4)
                   or (the vrbl = "e"
                       and the nchoices ~ < 5)
                    or (the vrbl = "f"
                          and the nchoices ~ < 6)),
                    forget about the question
                          and let the nchoices be 0
                    and return,
             otherwise if the vrbl = "-",
                    go count-unknowns
                    and forget about the question
                    and return,
             otherwise if the vrbl = "?",
                    go explain-quests,
             otherwise if the vrbl = "??",
                       send {1 line,
                            "I thought I told you that option is not ready yet...",
                                   1 line,":  "},
             otherwise if the vrbl = "q",
                    assert analysis is stopped
                    and forget about the question
                          and go wrap-up,
                otherwise,
                        send {1 line,"Improper form...PLEASE get it right..",
                                   1 line,":  "}.
```

End.

To input-before-after:

Execute cyclically.

```
[1] Send {":  "}
         and read {{0 or more blanks,
                         1 or more nonblanks (bind the string),cr}|
                         cr (bind the string)}
         and if (the lowercase of the string = "before"
                 or the lowercase of the string = "after"),
               forget about the question
               and return,
            otherwise if the string = "-",
               go count-unknowns
               and forget about the question
```

```
        and return,
      otherwise if the string = "?",
          go explain-quests,
      otherwise if the string = "??",
          send {1 line,
              "I thought I told you that option is not ready yet...",
                  1 line,": "},
      otherwise if the lowercase of the string = "q",
        assert analysis is stopped
          and forget about the question
        and go wrap-up,
      otherwise,
        send {1 line,
              "Improper answer, just Before/After/- PLEASE!",cr}.

End.

[**************************** Validity of entry. ****************************]

To decide entry is valid:

[1] Match the lowercase of the entry:
            {"january", anything}           conclude true;
            {"february", anything}          conclude true;
            {"march", anything}             conclude true;
            {"april", anything}             conclude true;
            {"may", anything}               conclude true;
            {"june", anything}              conclude true;
            {"july", anything}              conclude true;
            {"august", anything}            conclude true;
            {"september", anything}         conclude true;
            {"october", anything}           conclude true;
            {"november", anything}          conclude true;
            {"december", anything}          conclude true;

            default:                        conclude false.

End.

To decide answer is valid-answer:

[1] Match the lowercase of the answer:

          {"act-of-god"}               conclude true;
          {"act-of-third-party"}       conclude true;
          {"act-of-government"}        conclude true;
          {"act-of-contractor"}        conclude true;
          {"fault-of-contractor"}      conclude true;
          {"fault-of-government"}      conclude true;
          {"no-one"}                   conclude true;
```

```
        default:                 conclude false.

End.

To decide option is valid-option:

[1] Match the lowercase of the option:

        {"dsc assertion."}                  conclude true;
        {"final payment."}                  conclude true;
        {"notice form."}                    conclude true;
        {"notice promptness."}              conclude true;
        {"responsible receiver."}           conclude true;
        {"government prejudiced."}          conclude true;
        {"contract obligation."}            conclude true;
        {"excluded conditions."}            conclude true;
        {"after-bid conditions."}           conclude true;
        {"express-implied conditions."}     conclude true;
        {"latent deficiencies."}            conclude true;
        {"superior knowledge-i."}           conclude true;
        {"site inspection."}                conclude true;
        {"reliance i."}                     conclude true;
        {"material difference-i."}          conclude true;
        {"exculpatory language-i."}         conclude true;
        {"contract indications."}           conclude true;
        {"standard conditions."}            conclude true;
        {"superior knowledge-ii."}          conclude true;
        {"reliance ii."}                    conclude true;
        {"material difference-ii."}         conclude true;
        {"exculpatory language-ii."}        conclude true;
        {"complete analysis."}              conclude true;
        {"no module."}                      conclude true;

        default:                            conclude false.

End.

[****** Save answers to the questions asked during this cycle/session. ******]

To examine-results:

Execute cyclically.

[1] Send {1 line,5 blanks,"a.  The answers given to the questions      ",cr,
        5 blanks,"b.  The unknown information                 ",cr,
        5 blanks,"c.  The assumptions which have been made    ",cr,
        5 blanks,"d.  The conclusions which have been reached ",cr,
        5 blanks,"e.  All of the above                        ",cr,
        5 blanks,"f.  None of the above                       ",cr,
        ":  "}
     and read {{0 or more blanks,
```

```
                              1 or more nonblanks (bind the reply),cr}|
                              cr (bind the reply)}.

[2] If the reply = "a",
        send {1 line,"The answers which you have given to the ",cr,
          "questions are as follows:",cr}
        and activate answers
        and evaluate {"?"}
        and deactivate,
      otherwise if the reply = "b",
        send {1 line,
          "The items which are still unknown are:",cr}
        and activate unknowns
        and evaluate {"?"}
        and deactivate,
      otherwise if the reply = "c",
        send {1 line,"The following assumptions have been made:",cr}
        and activate assumptions
        and evaluate {"?"}
        and deactivate,
      otherwise if the reply = "d",
        send {1 line,
          "The following conclusions have been reached:",cr}
        and activate reports
        and evaluate {"?"}
        and deactivate,
      otherwise if the reply = "e",
        send {1 line,"These are the results of this session:",cr,
              1 line,"The answers:",cr}
        and activate answers
        and evaluate {"?"}
        and deactivate
        and send {1 line,"The unknowns:",cr}
        and activate unknowns
        and evaluate {"?"}
        and deactivate
        and send {1 line,"The assumptions:",cr}
        and activate assumptions
        and evaluate {"?"}
        and deactivate
        and send {1 line,"The conclusions:",cr}
        and activate reports
        and evaluate {"?"}
        and deactivate,
      otherwise if the reply = "f",
        return,
      otherwise if the reply = "?",
        send {2 lines,"Come on... This question is straight forward!!",1 line},
      otherwise,
        send {1 line,"Your choices are a-f ONLY!",cr}.

[3] Send {1 line,"Which of the other results do you want to see?",cr}.
```

End.

To store-cycle:

```
[1] Send {1 line,"Please enter your name or initials.  (Please place a '-'",cr,
          "between initials and/or name.)",cr,":  "}
        and read {0 or more blanks,
                      1 or more nonblanks (bind the author),cr}.

[2] If there is a current-session,
        evaluate {"go add '",the author," is an author-of-",
                              the current-session,"' to status."}
        and match (the current_date) against
                  {1 or more characters (bind the date),
                              1 blank,anything}
        and evaluate {"go add '",the date," is a date-of-",
                              the current-session,"' to status."}
        and forget about the date
        and evaluate {"go add '",the current-session,
                              " is a new-database' to answers."}
        and evaluate {"go add '",the current-session,
                              " is a new-database' to unknowns."}
        and evaluate {"delete each of ",the current-session,
                              ".database and u-",the current-session,
                                                  ".database."},
      otherwise,
        evaluate {"go add '",the author," is an author-of-",
                              the session-name,"' to status."}
        and match (the current_date) against
                  {1 or more characters (bind the date),
                              1 blank,anything}
        and evaluate {"go add '",the date," is a date-of-",
                              the session-name,"' to status."}
        and forget about the date
        and evaluate {"go add '",the session-name,
                              " is a new-database' to answers."}
        and evaluate {"go add '",the session-name,
                              " is a new-database' to unknowns."}
           and evaluate {"Go add '",the session-name,
                      " is a previous-session' to status."}.

[3] Activate answers
        and evaluate {"dump as ",the new-database,"."}
        and forget about the new-database
        and deactivate
        and activate unknowns
        and evaluate {"dump as u-",the new-database,"."}
        and forget about the new-database
        and deactivate.

[4] Delete status.database
        and activate status
```

```
        and dump as status
        and deactivate.

End.


[************ Load answers from previous session if requested. *************]

To load-session:

Execute cyclically.

[1] Send {1 line,"Which of the following sessions do you wish to continue?",cr,
             "(Please select only ONE, enter it EXACTLY as it appears.)",cr}
        and activate status
        and display every previous-session
        and send {1 line,":  "}
        and read {{0 or more blanks,
                     1 or more nonblanks (bind the rply),cr}|
                     cr (bind the rply)}
        and evaluate {"Assert ",the rply," is a current-session."}.

[2] If there is a current-session which is a previous-session,
        evaluate {"go add '",the current-session,
                     " is a current-session' to global."}
        and evaluate {"go add '",the current-session,
                     " is a current-session' to answers."}
        and evaluate {"go add '",the current-session,
                     " is a current-session' to unknowns."}
        and deactivate
        and activate answers
        and evaluate {"restore ",the current-session,"."}
        and forget about the new-database
        and deactivate
        and activate unknowns
        and evaluate {"restore u-",the current-session,"."}
        and forget about the new-database
        and deactivate
        and return,
      otherwise,
        send {1 line,"That is not a valid previous session.",cr,
                          "Please try again...",cr}
        and forget about the current-session.

End.


To name-session:

Execute Cyclically.

[1] Send {2 lines,"In order to allow future reference to this session ",cr,
                "it is necessary to name it. Please assign a name ",cr,
```

```
                "of 7 characters or less.",cr,
                "(The name may be alpha-numeric if desired.)",cr,":  "}
         and read {{0 or more blanks,
                7 or less nonblanks (bind the session-name),cr}|
                8 or more nonblanks (bind the bad-entry),cr|
                cr (bind the bad-entry)}.

[2] If there is a bad-entry,
         send {1 line,"READ -- 7 (Seven) characters or LESS...",
                            2 lines},
       otherwise,
         return.

End.


To get-option:

Execute cyclically.

[1] Read {{0 or more blanks, something (bind the option),cr}|
                   cr (bind the option)}
         and if the option is valid-option,
               match (the option) against {anything (bind the option-1),
                              1 blank,anything (bind the option-2),"."}
               and evaluate {"Assert ",the option-1,"-",the option-2,
                        " is a module."}
            and forget about each of option-1, option-2 and option
            and return,
          otherwise,
            send {1 line,"Improper form....  Please try again.",cr,":  "}.

End.


[************************ Explanation Rulesets. ************************]

To explain-entitlement:

[1] Execute Cyclically.

[1] Read {anything,"If ",something (bind the conclusion),cr|
            anything," or ",something (bind the conclusion),cr}
            from entitle.text
       and evaluate {"If ",the conclusion,", assert conclusion is correct."}.

[2] If conclusion is correct,
         send {"the ",the conclusion,".",cr}
         and deny conclusion is correct
         and close entitle.text
         and open (the file_to_be_read) to read
         and go read_from_file
         and forget about the conclusion
```

```
        and close everything
        and return.

End.


To read_from_file:

[1] Execute Cyclically.

[1] Read {anything,"If ",something (bind the rule),","}
            from the file_to_be_read
            and evaluate {"If ",the rule,", assert rule is correct."}.

[2] If rule is correct,
          deny rule is correct
          and close the file_to_be_read
          and send {2 lines,"The following rule proved to be true causing me",cr,
                "to draw the above conclusion:",2 lines,"If ",the rule,cr,
                  8 blanks, "then the ",the conclusion,".",cr}
          and send {2 lines,"The following statements contained by the rule",cr,
                    "are true:",2 lines}
           and let the tstate be 1
          and go find-true-statements
          and forget about the rule
           and forget about the tstate
             and return,
      otherwise,
          read {anything,"."} from the file_to_be_read.

End.


To find-true-statements:

[1] Execute Cyclically.

[1] Match (the rule) against {anything,"'",
                something (bind the true_statement),"'",
                anything (bind the rule), end}.

[2] Match (the rule) against {anything,"is true in ",
                7 or more letters (bind the dbase),anything}.

[3] If the dbase = "assumpt",
      evaluate {"If '",the true_statement,"' is true in assumptions,
      assert statement does test-true."},
    otherwise,
      evaluate {"If '",the true_statement,"' is true in ",the dbase,",
                assert statement does test-true."}.

[4] If statement does test-true,
     if the dbase = "assumpt",
```

```
    send {"[",the tstate,"]  ",the true_statement,
         "   (in assumptions)",2 lines}
    and increment 'the tstate'
    and deny statement does test-true,
  otherwise,
     send {"[",the tstate,"]  ",the true_statement,"   (in ",
                              the dbase,")",2 lines}
     and increment 'the tstate'
     and deny statement does test-true.

[5] Forget about the true_statement
      and forget about the dbase.

[6] If (the rule) is matched by {anything,"'",something,"'",anything},
         do nothing,
      otherwise,
         return.

End.


[1] To generate a file_to_be_read:

[1] [1] Match the conclusion:

        {"contractor did probably concede rights-to-claim"}
                                     produce conclude1.text;
        {"contract did assign obligation-risk"}
                                     produce conclude1.text;
        {"condition is not considered to-be DSC"}
                                     produce conclude1.text;
        {"contractor did contribute to condition"}
                                     produce conclude1.text;
        {"condition is excluded condition"}      produce conclude2.text;
        {"condition is best claimed through var-in-est-quant-clause"}
                                     produce conclude2.text;
        {"contractor did fail to-heed contract indications"}
                                     produce conclude3.text;
        {"contractor is not able-to establish standard-of-expectation"}
                                     produce conclude3.text;
        {"condition does not differ from standard-of-expectation"}
                                     produce conclude3.text;
        {"contractor did have imputed knowledge-of-condition"}
                                     produce conclude4.text;
        {"contractor did have superior knowledge"}
                                     produce conclude4.text;
        {"contractor did not make simple inquiries"}
                                     produce conclude4.text;
        {"reasonable inspection is required-for-entitlement"}
                                     produce conclude4.text;
        {"contractor bid did not reflect condition"}
                                     produce conclude5.text;
        {"difference was not material"}          produce conclude5.text;
```

```
                    {"exculpatory-clause is probably valid"}
                                        produce conclude5.text.

End.


[*********************** Miscellaneous rulesets. ************************]

To create-dbases:

[1] Activate reports and deactivate
        and activate answers and deactivate
        and activate unknowns and deactivate
        and activate assumptions and deactivate
        and assert dbases are created.

End.


To clean-up:

[1] Clear answers
          and clear unknowns
          and clear assumptions
          and clear reports
          and deactivate
          and clear database
          and assert files are loaded.

End.


To count-unknowns:

[1] Let the ucount be (the ucount + 1).

[2] If the ucount = 11,
          send {2 lines,"You have answered 11 questions with unknown (-),",cr,
                    "1 more and I will be forced to stop...!",2 lines},
      otherwise if the ucount = 12,
          send {2 lines,"I'm sorry but you seem to be quite ill-informed.",cr,
                    "Further analysis would prove to be fruitless.",2 lines,
                    "I must quit.",2 lines,
                    "You have stated the following unknowns:",cr}
          and activate unknowns
          and evaluate ("?")
          and deactivate
          and go wrap-up.

End.


To locate-place:
```

[1] Choose situation:

If exculpatory-language-I was checked,
  deny exculpatory-language-I was checked;
If exculpatory-language-II was checked,
  deny exculpatory-language-II was checked;
If material-difference-I was checked,
  deny material-difference-I was checked;
If material-difference-II was checked,
  deny material-difference-II was checked;
If reliance-I was checked,
  deny reliance-I was checked;
If reliance-II was checked,
  deny reliance-II was checked;
If site-inspection was checked,
  deny site-inspection was checked;
If knowledge-I was checked,
  deny knowledge-I was checked;
If knowledge-II was checked,
  deny knowledge-II was checked;
If standard-of-comparison was checked,
  deny standard-of-comparison was checked;
If latent-deficiencies was checked,
  deny latent-deficiencies was checked;
If contract-indications was checked,
  deny contract-indications was checked;
If express-implied-deficiencies was checked,
  deny express-implied-deficiencies was checked;
If after-bid-conditions was checked,
  assert cause-of-condition is unknown
  and deny after-bid-conditions was checked;
If excluded-conditions was checked,
  deny excluded-conditions was checked;
If obligation was checked,
  deny obligation was checked;
If prejudice was checked,
  deny prejudice was checked;
If responsible-receiver was checked,
  deny responsible-receiver was checked;
If promptness was checked,
  deny promptness was checked;
If form-of-notice was checked,
  deny form-of-notice was checked;
If final-payment was checked,
  deny final-payment was checked;
If dsc-assertion was checked,
  deny dsc-assertion was checked.

End.

To file-cleaner:

```
Execute Cyclically.

[1] Send {1 line,"Which ONE of the following files do you ",cr,
                "wish to obliterate? ",cr}
        and activate status
        and display every previous-session
        and send {1 line,": "}
        and read {{something (bind the rply),cr}| cr (bind the rply)}
        and evaluate {"Assert ",the rply," is a filen."}
        and forget about rply.

[2] If there is a filen which is a previous-session,
        evaluate {"delete each of ",the filen,",",the filen,
                ".database and u-",the filen,".database."}
        and evaluate {"Forget about each of date-of-",the filen,
                " and the author-of-",the filen,"."}
        and forget about the filen
        and deactivate
        and send {"Any others? : "}
        and assert ss is a question
        and go read-yes-no
        and (if the lowercase of the string = "yes",
                        do nothing,
             otherwise quit),
      otherwise,
        send {1 line,"Not a previous session. (Dingbat!!!)",cr}
        and deactivate.

End.
```

APPENDIX C

TEST CASES

73-2 BCA

[¶ 10,309] ALPS CONSTRUCTION CORPORATION

ASBCA No. 16966. September 17, 1973.

Contract No. DACA 41-69-C-0074.

Changed Conditions--Rock and Stone--Reliability of Government-furnished Information.

A contractor was entitled to an equitable adjustment for removal of unexpected hard rock encountered in excavating for a well water system because he reasonably interpreted information from government borings to indicate that he would encounter only small rocks capable of being excavated efficiently with the equipment he intended to use. The fact that he found some rock in the government's test holes did not require him to provide a contingency in his bid to cover the possibility that hard rock might be encountered, nor should he have realized that the rock samples were merely chips from larger underground boulders. The government had information in its field logs to indicate the presence of large boulders, but it neither attached the field logs to the bidding documents nor told bidders where they could be seen. The contractor's failure to make a site inspection was not detrimental to his claim because the conditions encountered would not have been apparent.

TM
[Rosie Version 2.3 5-DEC-83 01:28:36]

The name of this session shall be: ALPS

You may perform any one of the following analyses:

DSC Assertion Final Payment Notice Form
Notice Promptness Responsible Receiver Government Prejudiced

Contract Obligation Excluded Conditions After-bid Conditions

Express-implied Conditions Contract Indications
Latent Deficiencies Standard Conditions
Site Inspection

Superior Knowledge-I Superior Knowledge-II
Reliance I Reliance II
Material Difference-I Material Difference-II
Exculpatory Langauge-I Exculpatory Language-II

Complete Analysis No Module

I will begin the analysis as soon as you have selected any ONE
of the options listed. Please enter the selected option
EXACTLY as it appears in the list followed by a '.'.
: Complete Analyssis.

SESSION IN PROGRESS

[1] Has a differing site condition been asserted by one of
your contractors? (Yes/No)
: Yes

[2] On what date did the DSC-ASSERTION occur? (month day, year)
: October 1, 1969

[3] Has final payment been made? (Yes/No)
: No

[4] What form of notice was given to the government by
the contractor? (Select one of the following.)
a. Written notice was given.
b. Oral notice was given.
c. Other events surrounded notice.

: a

[5] On what date did the WRITTEN-NOTICE occur? (month day, year)
: September 11, 1969

[6] When was notice given in relation to the progression of work?
a. Notice given before contractor disturbed the condition.
b. Notice given before contractor performed remedial work.
c. Both a and b.
d. None of the above.
: c

[7] Is there evidence that the contractor should have been aware of the differing site condition sooner? (Yes/No)
: No

[8] To whom was notice of the differing site condition given?
a. The contracting officer
b. An authorized representative of the contracting officer
c. Any other employee of the government
d. None of the above
: a

[9] Who expressly assumed obligation for or risk of the condition in the contract?
a. The government
b. The contractor
c. Both the government and the contractor
d. Neither the government nor the contractor
: -

[10] Is the condition in question directly related to the physical conditions at the work site? (Yes/No)
: Yes

[11] Is the condition a static physical part of the work site? (Yes/No)
: Yes

[12] Did the condition occur before or after the contract award?
: Before

[13] Were there affirmatively expressed statements in the
contract concerning the relevant subsurface or
latent conditions at the site? (Yes/No)
: Yes

There were indications regarding the condition
in the contract, therefore, a Type I condition is
the most probable avenue of recovery.

[14] Were there latent deficiencies in the contract
indications? (Yes/No)
: Yes

[15] Was the physical data made a part of the
contract? (Yes/No)
: Yes

[16] Would this physical data have revealed or resolved
the apparent deficiencies in the contract indications for
a reasonable contractor? (Yes/No)
: No

[17] Did the GOVERNMENT know about the
CONTRACT-DEFICIENCY? (Yes/No)
: Yes

[18] Did the government reveal CONTRACT-DEFICIENCY to the contractor?
(Yes/No)
: No

Oversight, misrepresentation or concealment on
the part of the government. A possible
breach has occurred. However, I will
continue with the analysis.

[19] Was the contractor in possession of unilateral superior
or imputed knowledge that the actual conditions patently
differed from the indications in the contract? (Yes/No)
: No

[20] Would simple inquiries by the contractor have revealed the condition to be contrary to the CONTRACT-INDICATIONS? (Yes/No)
: -

[21] Did any of the following acts of the government hinder the site inspection?

a. Access to site was denied
b. Inadequate time was allowed
c. No act of the government hindered inspection
: c

[22] Did the contractor conduct a site inspection? (Yes/No)
: No

Your contractor risked encountering an unexpected condition that a reasonable inspection might have revealed.

[23] Would the actual condition have been discernable by a layman contractor performing a reasonable inspection? (Yes/No)
: No

[24] Did the contractor reasonably rely on DEFICIENT-INDICATIONS-IN-CONTRACT when preparing his bid causing him to suffer prejudice through this reliance? (Yes/No)
: Yes

[25] Was the difference between actual and INDICATED conditions a material difference? (Yes/No)
: Yes

Relief is probably available through the Type I DSC clause because of deficient indications in the contract.

[26] Is there an exculpatory clause denying any government liability and responsibility for actual conditions

different from those indicated in the contract
documents? (Yes/No)
: Yes

[27] Is the exculpatory clause specific to the DSC clause and to the condition encountered? (Yes/No)
: No

[28] Are the language and intent of the exculpatory clause clear and unambiguous? (Yes/No)
: No

I believe that the exculpatory clause is
probably not valid for this case.

Therefore, entitlement will probably be allowed.
Consult your local CO for further details.

Thinking.......

********************* END OF ANALYSIS *********************
(You should receive my bill in 7-10 days.)
(PROMPT PAYMENT IS APPRECIATED)

Do you wish to see the results from this session? (Yes/No)
: Yes

Which of the following results do you wish
to see?

a. The answers given to the questions
b. The unknown information
c. The assumptions which have been made
d. The conclusions which have been reached
e. All of the above
f. None of the above

: e

These are the results of this session:

The answers:
[ANSWERS Database]
CONDITION did occur before CONTRACT AWARD.
CONTRACTOR did not have POTENTIAL-TO-KNOW CONDITION SOONER.

REASONABLE INSPECTION did not have POTENTIAL-TO-REVEAL CONDITION.
CONTRACT did contain STATEMENTS CONCERNING CONDITION.
CONTRACT INDICATIONS did contain LATENT DEFICIENCIES.
CONTRACT INFORMATION did contain PHYSICAL DATA.
CONTRACT INFORMATION did not reveal DEFICIENCIES/CONDITIONS.
GOVERNMENT did not reveal CONTRACT-DEFICIENCY.
GOVERNMENT did not hinder INSPECTION.
CONTRACTOR did not conduct SITE-INSPECTION.
CONTRACTOR did suffer PREJUDICE through RELIANCE.
CONTRACT does contain EXCULPATORY-CLAUSE.
<OCTOBER 1, 1969> is a date-of-dsc-assertion.
<SEPTEMBER 11, 1969> is a date-of-written-notice.
NOTICE was received-by CONTRACTING OFFICER.
GOVERNMENT was aware-of CONTRACT-DEFICIENCY.
CONTRACTOR was not aware-of CONTRACT-DEFICIENCY.
CONDITION is static PHYSICAL PART OF-WORK-SITE.
EXCULPATORY-CLAUSE is not specific-to DSC CLAUSE.
DSC was asserted.
FINAL-PAYMENT was not made.
FORM-OF-NOTICE was written.
NOTICE was given before CONTRACTOR DISTURBED CONDITION.
NOTICE was given before REMEDIAL WORK PERFORMED.
DIFFERENCE was material.
CONDITION is directly-related-to-physical-conditions.
EXCULPATORY-CLAUSE is not clear-and-unambiguous.

The unknowns:
[UNKNOWNS Database]
WHICH-PARTY did assume RISK.
IF-SIMPLE INQUIRY did give POTENTIAL TO-KNOW CONTRARY CONDITIONS.

The assumptions:
[ASSUMPTIONS Database]
GOVERNMENT did not expressly ASSUME OBLIGATION-RISK.
CONTRACTOR did not expressly ASSUME OBLIGATION-RISK.
SIMPLE INQUIRY did not give POTENTIAL TO-KNOW CONTRARY CONDITIONS.

The conclusions:
[REPORTS Database]
CONTRACTOR did comply with WRITTEN-NOTICE-REQUIREMENT.
CONTRACTOR did comply-fully with NOTICE-REQUIREMENT.
GOVERNMENT did receive NOTICE-OF-CONDITION.
CONTRACT did contain INDICATIONS CONCERNING CONDITION.
GOVERNMENT did have SUPERIOR KNOWLEDGE.
CONTRACTOR did not conduct SITE-INSPECTION.
NOTICE was given-to RESPONSIBLE RECEIVER.
CONTRACTOR is not required-to INSPECT OTHER DATA.
EXCULPATORY-CLAUSE is not probably VALID.
NOTICE was prompt.
REASONABLE INSPECTION is not required-for-entitlement.

VALIDITY-OF EXCULPATORY-CLAUSE is in_doubt.
ENTITLEMENT is still-probable.

Which of the other results do you want to see?

a. The answers given to the questions
b. The unknown information
c. The assumptions which have been made
d. The conclusions which have been reached
e. All of the above
f. None of the above

: f

Do you wish to store this session? (Yes/No)
: Yes

Please enter your name or initials. (Please place a '-' between initials and/or name.)
: TaK

Session has been saved as requested.

Negotiation Memorandum

Contract No. 8-07-DC-B7332
Modification No. 5 (Part 2 of a 3-part)

On March 7, 1979, the contractor began excavation at Station 915 + 69 to uncover a 24-inch sewerline in which he encountered water 8 feet below the surface. On March 15, 1970, the contractor excavated three test pits at Stations 919 + 50, and 940 + 00, in which he encountered water at 18, 5, and 10 feet below the surface respectively. By letter dated March 22, 1979, the contractor notified the Government that they had found water at elevations above the top of installed pipe between Stations 915 + 50 and 926 + 00. With the encountering of ground water the contractor initiated a dewatering program to lower the water table. On April 20, 1979, the contractor began excavating for the pipe trench at Station 915 + 50. Excavation was slow due to the ground water and instability of the trench walls. By letter dated May 9, 1979, the contractor notified the Government that he was being delayed as a result of the ground water and unstable material. Excavation continued until weekend of May 20, 1979, when a large amount of sloughing occurred in the trench walls. On May 29, 1979, a major amount of sloughing occurred between Stations 922 + 35 and 922 + 80, this sloughing was causing the shoring to scissor and collapse at which time the contractor elected to pull his men out of the trench and backfill it due to the unsafe and hazardous conditions. By letter dated May 30,

1979, the contractor stated that the unforeseen presence of significant ground water was increasing their construction costs and scheduled production and that they will be submitting for our review and approval a claim for additional contract time and costs upon completion of their analysis.

TM
[Rosie Version 2.3 5-DEC-83 02:31:55]

The name of this session shall be: BUR-REC

You may perform any one of the following analyses:

DSC Assertion   Final Payment   Notice Form
Notice Promptness   Responsible Receiver   Government Prejudiced

Contract Obligation   Excluded Conditions   After-bid Conditions

Express-implied Conditions   Contract Indications
Latent Deficiencies   Standard Conditions
Site Inspection

Superior Knowledge-I   Superior Knowledge-II
Reliance I   Reliance II
Material Difference-I   Material Difference-II
Exculpatory Langauge-I   Exculpatory Language-II

Complete Analysis   No Module

I will begin the analysis as soon as you have selected any ONE
of the options listed. Please enter the selected option
EXACTLY as it appears in the list followed by a '.'.
: Complete Analysis.

SESSION IN PROGRESS

[1] Has a differing site condition been asserted by one of
your contractors? (Yes/No)
: Yes

[2] On what date did the DSC-ASSERTION occur? (month day, year)
: May 30, 1979

[3] Has final payment been made? (Yes/No)
: No

[4] What form of notice was given to the government by
the contractor? (Select one of the following.)

a. Written notice was given.
b. Oral notice was given.
c. Other events surrounded notice.

: a

[5] On what date did the WRITTEN-NOTICE occur? (month day, year)
: March 22, 1979

[6] When was notice given in relation to the progression of work?
a. Notice given before contractor disturbed the condition.
b. Notice given before contractor performed remedial work.
c. Both a and b.
d. None of the above.
: c

[7] Is there evidence that the contractor should have been aware of the differing site condition sooner? (Yes/No)
: No'

[8] To whom was notice of the differing site condition given?
a. The contracting officer
b. An authorized representative of the contracting officer
c. Any other employee of the government
d. None of the above
: a

[9] Who expressly assumed obligation for or risk of the condition in the contract?
a. The government
b. The contractor
c. Both the government and the contractor
d. Neither the government nor the contractor
: d

[10] Is the condition in question directly related to the physical conditions at the work site? (Yes/No)
: Yes

[11] Is the condition a static physical part of the work site? (Yes/No)
: Yes

[12] Did the condition occur before or after the contract award?
: Before

[13] Were there affirmatively expressed statements in the contract concerning the relevant subsurface or latent conditions at the site? (Yes/No)
: -

[14] Would the affirmatively expressed contract statements on the general conditions at the site lead a reasonable contractor to believe that the condition could be expected? (Yes/No)
: -

[15] Were indications of subsurface or latent conditions inferred from the contract as a whole? (Yes/No)
: Yes

[16] Was this inference justified based on the information in the contract? (Yes/No)
: Yes

There were indications regarding the condition in the contract, therefore, a Type I condition is the most probable avenue of recovery.

[17] Were there latent deficiencies in the contract indications? (Yes/No)
: -

[18] Was the physical data made a part of the contract? (Yes/No)
: -

[19] Did the GOVERNMENT know about the CONTRACT-DEFICIENCY? (Yes/No)
: No

[20] Was the contractor in possession of unilateral superior or imputed knowledge that the actual conditions patently differed from the indications in the contract? (Yes/No)
: No

[21] Would simple inquiries by the contractor have revealed the condition to be contrary to the CONTRACT-INDICATIONS? (Yes/No)
: No

[22] Did any of the following acts of the government hinder the site inspection?

a. Access to site was denied
b. Inadequate time was allowed
c. No act of the government hindered inspection
: c

[23] Did the contractor conduct a site inspection? (Yes/No)
: Yes

[24] Was the inspection conducted by the contractor a reasonable inspection, i.e., one comparable to that expected from a reasonable, prudent contractor experienced in that particular field of work? (Yes/No)
: Yes

[25] Would the actual condition have been discernable by a layman contractor performing a reasonable inspection? (Yes/No)
: No

[26] Did the contractor reasonably rely on DEFICIENT-INDICATIONS-IN-CONTRACT when preparing his bid causing him to suffer prejudice through this reliance? (Yes/No)
: Yes

[27] Was the difference between actual and INDICATED conditions a material difference? (Yes/No)
: Yes

Relief is probably available through the Type I DSC clause because of deficient indications in the contract.

[28] Is there an exculpatory clause denying any government
liability and responsibility for actual conditions
different from those indicated in the contract
documents? (Yes/No)
: -

I have assumed that the contract does not contain
an exculpatory clause. Therefore, entitlement will
probably be allowed.

Consult your local CO for further details.

Thinking.......

********************* END OF ANALYSIS *********************
(You should receive my bill in 7-10 days.)
(PROMPT PAYMENT IS APPRECIATED)

Do you wish to see the results from this session? (Yes/No)
: Yes

Which of the following results do you wish
to see?

a. The answers given to the questions
b. The unknown information
c. The assumptions which have been made
d. The conclusions which have been reached
e. All of the above
f. None of the above

: e

These are the results of this session:

The answers:
[ANSWERS Database]
CONDITION did occur before CONTRACT AWARD.
CONTRACTOR did not have POTENTIAL-TO-KNOW CONDITION SOONER.
REASONABLE INSPECTION did not have POTENTIAL-TO-REVEAL CONDITION.
GOVERNMENT did not expressly ASSUME OBLIGATION-RISK.
CONTRACTOR did not expressly ASSUME OBLIGATION-RISK.
CONTRACT did lead ONE-TO INFER CONDITIONS.
CONTRACT INFORMATION did justify INFERENCE.
SIMPLE INQUIRY did not give POTENTIAL TO-KNOW CONTRARY CONDITIONS.
GOVERNMENT did not hinder INSPECTION.
CONTRACTOR did conduct SITE-INSPECTION.
CONTRACTOR did suffer PREJUDICE through RELIANCE.
<MAY 30, 1979> is a date-of-dsc-assertion.

<MARCH 22, 1979> is a date-of-written-notice.
NOTICE was received-by CONTRACTING OFFICER.
GOVERNMENT was not aware-of CONTRACT-DEFICIENCY.
CONTRACTOR was not aware-of CONTRACT-DEFICIENCY.
SITE-INSPECTION was reasonable INSPECTION.
CONDITION is static PHYSICAL PART OF-WORK-SITE.
DSC was asserted.
FINAL-PAYMENT was not made.
FORM-OF-NOTICE was written.
NOTICE was given before CONTRACTOR DISTURBED CONDITION.}i
NOTICE was given before REMEDIAL WORK PERFORMED.
DIFFERENCE was material.
CONDITION is directly-related-to-physical-conditions.

The unknowns:
[UNKNOWNS Database]
IF-CONTRACT did contain STATEMENTS CONCERNING CONDITION.
IF-CONTRACT did contain GENERAL CONDITIONS IMPLYING CONDITION.
IF-CONTRACT INDICATIONS did contain LATENT DEFICIENCIES.
IF-CONTRACT INFORMATION did contain PHYSICAL DATA.
IF-CONTRACT does contain EXCULPATORY-CLAUSE.

The assumptions:
[ASSUMPTIONS Database]
CONTRACT did not contain STATEMENTS CONCERNING CONDITION.
CONTRACT did not contain GENERAL CONDITIONS IMPLYING CONDITION.
CONTRACT INDICATIONS did contain LATENT DEFICIENCIES.
CONTRACT INFORMATION did not contain PHYSICAL DATA.
CONTRACT does not contain EXCULPATORY-CLAUSE.

The conclusions:
[REPORTS Database]
CONTRACTOR did comply with WRITTEN-NOTICE-REQUIREMENT.
CONTRACTOR did comply-fully with NOTICE-REQUIREMENT.
GOVERNMENT did receive NOTICE-OF-CONDITION.
CONTRACT did not assign OBLIGATION-RISK.
CONTRACT did contain INDICATIONS CONCERNING CONDITION.
REASONABLE INSPECTION did not reveal CONDITION.
NOTICE was given-to RESPONSIBLE RECEIVER.
NOTICE was prompt.
REASONABLE INSPECTION is not required-for-entitlement.
ENTITLEMENT is still-probable.

Which of the other results do you want to see?

a. The answers given to the questions
b. The unknown information
c. The assumptions which have been made
d. The conclusions which have been reached

e. All of the above
f. None of the above
: f

Do you wish to store this session? (Yes/No)
: Yes

Please enter your name or initials. (Please place a '-' between initials and/or name.)
: TaK

Session has been saved as requested.

75-2 BCA

[¶ 11,541] CONTINENTAL DRILLING COMPANY

ENG BCA No. 3455. September 9, 1975.

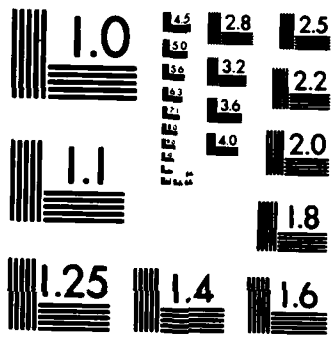
Contract No. DACW07-71-C-002.

Changed conditions--Rock, Water and Other Subsurface Conditions--Latent Conditions.

A core drilling contractor was entitled to an equitable adjustment for extra work caused by unforeseeable cave-ins because the subsurfact conditions at the site differed materially from those indicated in the contract. The contract led the contractor to believe that the subsurface conditions he would encounter at the drilling site would be firm, well cemented soil since the contract only called for 30 linear feet of casing. However, he encountered a very substantial amount of material that was not well cemented and tight in place, but was susceptible to caving in as it was being drilled, thus requiring him to use 455 feet of casing. The contractor reasonably relied on the casing requirements and expected firm subsurface soil. His lack of site inspection and his failure to inspect government logs, reports, and core samples had no bearing on what he in fact encountered because they would have given him no hint regarding the actual subsurface conditions.

AD-A146 681 THE APPLICATION OF ARTIFICIAL INTELLIGENCE TO CONTRACT MANAGEMENT(U) CONSTRUCTION ENGINEERING RESEARCH LAB (ARMY) CHAMPAIGN IL T A KRUPPENBACHER AUG 84 5/6

UNCLASSIFIED CERL-TM-P-166 F/G 9/1 NL



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

TM
[Rosie Version 2.3 5-DEC-83 01:05:50]

The name of this session shall be: CONTINE

You may perform any one of the following analyses:

DSC Assertion   Final Payment   Notice Form
Notice Promptness   Responsible Receiver   Government Prejudiced

Contract Obligation   Excluded Conditions   After-bid Conditions

Express-implied Conditions   Contract Indications
Latent Deficiencies   Standard Conditions
Site Inspection

Superior Knowledge-I   Superior Knowledge-II
Reliance I   Reliance II
Material Difference-I   Material Difference-II
Exculpatory Langauge-I   Exculpatory Language-II

Complete Analysis   No Module

I will begin the analysis as soon as you have selected any ONE
of the options listed. Please enter the selected option
EXACTLY as it appears in the list followed by a '.'.
: Complete Analysis.

SESSION IN PROGRESS

[1] Has a differing site condition been asserted by one of
your contractors? (Yes/No)
: Yes

[2] On what date did the DSC-ASSERTION occur? (month day, year)
: -

[3] Has final payment been made? (Yes/No)
: No

[4] What form of notice was given to the government by
the contractor? (Select one of the following.)
a. Written notice was given.
b. Oral notice was given.
c. Other events surrounded notice.

: -

[5] When was notice given in relation to the progression of work?
a. Notice given before contractor disturbed the condition.
b. Notice given before contractor performed remedial work.
c. Both a and b.
d. None of the above.
: -

[6] Is there evidence that the contractor should have been aware of the differing site condition sooner? (Yes/No)
: No

[7] To whom was notice of the differing site condition given?
a. The contracting officer
b. An authorized representative of the contracting officer
c. Any other employee of the government
d. None of the above
: -

Receipt of notice by responsible government employee is in doubt.

[8] Is the government able to show that evidence against the claim was obscurred due to the passage of time? (Yes/No)
: No

[9] Is the government able to show that proper notice would have resulted in cheaper resolution of the condition by CO? (Yes/No)
: No

[10] Can the government show that it suffered any other prejudice through inadequacy of or lack of notice? (Yes/No)
: No

Government can't prove that it suffered prejudice.

[11] Who expressly assumed obligation for or risk of the condition in the contract?
a. The government

b. The contractor
c. Both the government and the contractor
d. Neither the government nor the contractor
: d

[12] Is the condition in question directly related to the physical conditions at the work site? (Yes/No)
: Yes

[13] Is the condition a static physical part of the work site? (Yes/No)
: Yes

[14] Did the condition occur before or after the contract award?
: Before

[15] Were there affirmatively expressed statements in the contract concerning the relevant subsurface or latent conditions at the site? (Yes/No)
: No

[16] Would the affirmatively expressed contract statements on the general conditions at the site lead a reasonable contractor to believe that the condition could be expected? (Yes/No)
: No

[17] Were indications of subsurface or latent conditions inferred from the contract as a whole? (Yes/No)
: Yes

[18] Was this inference justified based on the information in the contract? (Yes/No)
: Yes

There were indications regarding the condition in the contract, therefore, a Type I condition is the most probable avenue of recovery.

[19] Were there latent deficiencies in the contract indications? (Yes/No)
: Yes

[20] Was the physical data made a part of the contract? (Yes/No)
: Yes

[21] Would this physical data have revealed or resolved the apparent deficiencies in the contract indications for a reasonable contractor? (Yes/No)
: No

[22] Did the GOVERNMENT know about the CONTRACT-DEFICIENCY? (Yes/No)
: -

[23] Was the contractor in possession of unilateral superior or imputed knowledge that the actual conditions patently differed from the indications in the contract? (Yes/No)
: No

[24] Would simple inquiries by the contractor have revealed the condition to be contrary to the CONTRACT-INDICATIONS? (Yes/No)
: No

[25] Did any of the following acts of the government hinder the site inspection?

a. Access to site was denied
b. Inadequate time was allowed
c. No act of the government hindered inspection
: c

[26] Did the contractor conduct a site inspection? (Yes/No)
: No

Your contractor risked encountering an unexpected condition that a reasonable inspection might have revealed.

[27] Would the actual condition have been discernable
by a layman contractor performing a reasonable
inspection? (Yes/No)
: No

[28] Did the contractor reasonably rely on DEFICIENT-INDICATIONS-IN-CONTRACT
when preparing his bid causing him to suffer prejudice
through this reliance? (Yes/No)
: Yes

[29] Was the difference between actual and INDICATED
conditions a material difference? (Yes/No)
: Yes

Relief is probably available through the Type I
DSC clause because of deficient indications in
the contract.

[30] Is there an exculpatory clause denying any government
liability and responsibility for actual conditions
different from those indicated in the contract
documents? (Yes/No)
: No

Since the contract does not contain an exculpatory
clause entitlement will probably be allowed.

Consult your local CO for further details.

Thinking.......

********************** END OF ANALYSIS **********************
(You should receive my bill in 7-10 days.)
(PROMPT PAYMENT IS APPRECIATED)

Do you wish to see the results from this session? (Yes/No)
: Yes

Which of the following res[illegible]s do you wish

to see?

a. The answers given to the questions
b. The unknown information
c. The assumptions which have been made
d. The conclusions which have been reached
e. All of the above
f. None of the above

: e

These are the results of this session:

The answers:
[ANSWERS Database]
CONDITION did occur before CONTRACT AWARD.
CONTRACTOR did not have POTENTIAL-TO-KNOW CONDITION SOONER.
REASONABLE INSPECTION did not have POTENTIAL-TO-REVEAL CONDITION.
PASSAGE-OF-TIME did not obscure EVIDENCE.
IMPROPER-NOTICE did not cause ADDITIONAL COST.
NOTHING-ELSE did cause PREJUDICE.
GOVERNMENT did not expressly ASSUME OBLIGATION-RISK.
CONTRACTOR did not expressly ASSUME OBLIGATION-RISK.
CONTRACT did not contain STATEMENTS CONCERNING CONDITION.
CONTRACT did not contain GENERAL CONDITIONS IMPLYING CONDITION.
CONTRACT INDICATIONS did contain LATENT DEFICIENCIES.
CONTRACT INFORMATION did contain PHYSICAL DATA.
CONTRACT did lead ONE-TO INFER CONDITIONS.
CONTRACT INFORMATION did justify INFERENCE.
CONTRACT INFORMATION did not reveal DEFICIENCIES/CONDITIONS.
SIMPLE INQUIRY did not give POTENTIAL TO-KNOW CONTRARY CONDITIONS.
GOVERNMENT did not hinder INSPECTION.
CONTRACTOR did not conduct SITE-INSPECTION.
CONTRACTOR did suffer PREJUDICE through RELIANCE.
CONTRACT does not contain EXCULPATORY-CLAUSE.
<UNKNOWN> is a date-of-dsc-assertion.
CONTRACTOR was not aware-of CONTRACT-DEFICIENCY.
CONDITION is static PHYSICAL PART OF-WORK-SITE.
DSC was asserted.
FINAL-PAYMENT was not made.
DIFFERENCE was material.
CONDITION is directly-related-to-physical-conditions.

The unknowns:
[UNKNOWNS Database]
TO-WHOM was notice GIVEN.
IF-GOVERNMENT was aware-of CONTRACT-DEFICIENCY.
IF-NOTICE was given before CONDITION DISTURBED.
FORM-OF-NOTICE is unknown.

The assumptions:
[ASSUMPTIONS Database]

NOTICE was not received-by GOVERNMENT.
GOVERNMENT was not aware-of CONTRACT-DEFICIENCY.
FORM-OF-NOTICE was other.
NOTICE was given before CONDITION DISTURBED.
NOTICE was given before REMEDIAL WORK PERFORMED.

The conclusions:
[REPORTS Database]
CONTRACTOR did constructively COMPLY with NOTICE-REQUIREMENT.
GOVERNMENT did not suffer PREJUDICE.
CONTRACT did not assign OBLIGATION-RISK.
CONTRACT did contain INDICATIONS CONCERNING CONDITION.
CONTRACTOR did not conduct SITE-INSPECTION.
RECEIPT-OF-NOTICE is an in_doubt.
CONTRACTOR is not required-to INSPECT OTHER DATA.
NOTICE was prompt.
REASONABLE INSPECTION is not required-for-entitlement.
ENTITLEMENT is still-probable.

Which of the other results do you want to see?

a. The answers given to the questions
b. The unknown information
c. The assumptions which have been made
d. The conclusions which have been reached
e. All of the above
f. None of the above

: f

Do you wish to store this session? (Yes/No)
: Yes

Please enter your name or initials. (Please place a '-'
between initials and/or name.)
: TaK

Session has been saved as requested.

77-1 BCA

[¶ 12,511] DE MAURO CONSTRUCTION CORPORATION

ASBCA No. 17029. April 28, 1977.

Contract No. CACA79-68-C-0032.

Changes--Notice of Changed Condition Existing--Opportunity for Government to Investigate.

The government was prejudiced by a contractor's failure to provide notice of his claim for unanticipated rock uncovered during an excavation for a water main because the material was dumped into the ocean where it was dispersed by wave action. The government, therefore, never had a chance to investigate the contractor's claim. The Changed Conditions clause of the contract required written notification of changed condition claims. The clause, however, permitted waiver of the notice requirement if notice were given before final payment. This notice could be waived if there was no prejudice to the government. Since the government could not investigate the accuracy of the contractor's claim and the contractor did not furnish the government with survey notes, the government was prejudiced and the claim was barred for lack of timely notice.

TM
[Rosie Version 2.3 3-DEC-83 23:52:55]

The name of this session shall be: DeMAURO

You may perform any one of the following analyses:

DSC Assertion Final Payment Notice Form
Notice Promptness Responsible Receiver Government Prejudiced

Contract Obligation Excluded Conditions After-bid Conditions

Express-implied Conditions Contract Indications
Latent Deficiencies Standard Conditions
Site Inspection

Superior Knowledge-I Superior Knowledge-II
Reliance I Reliance II
Material Difference-I Material Difference-II
Exculpatory Langauge-I Exculpatory Language-II

Complete Analysis No Module

I will begin the analysis as soon as you have selected any ONE of the options listed. Please enter the selected option EXACTLY as it appears in the list followed by a '.'.
: Complete Analysis.

SESSION IN PROGRESS

[1] Has a differing site condition been asserted by one of your contractors? (Yes/No)
: Yes

[2] On what date did the DSC-ASSERTION occur? (month day, year)
: July 26 , 1968

[3] Has final payment been made? (Yes/No)
: No

[4] What form of notice was given to the government by the contractor? (Select one of the following.)
a. Written notice was given.
b. Oral notice was given.
c. Other events surrounded notice.

: a

[5] On what date did the WRITTEN-NOTICE occur? (month day, year)
: July 26, 1968

[6] When was notice given in relation to the progression of work?
a. Notice given before contractor disturbed the condition.
b. Notice given before contractor performed remedial work.
c. Both a and b.
d. None of the above.
: d

[7] Is there evidence that the contractor should have been aware of the differing site condition sooner? (Yes/No)
: Yes

[8] To whom was notice of the differing site condition given?
a. The contracting officer
b. An authorized representative of the contracting officer
c. Any other employee of the government
d. None of the above
: a

[9] Is the government able to show that evidence against the claim was obscurred due to the passage of time? (Yes/No)
: Yes

[10] Is the government able to show that proper notice would have resulted in cheaper resolution of the condition by CO? (Yes/No)
: -

[11] Can the government show that it suffered any other prejudice through inadequacy of or lack of notice? (Yes/No)
: -

The government did suffer prejudice.

[12] Was the govenment defense made impossible by inadequacy of or lack of notice? (Yes/No)
: -

[13] Was the contractor able to provide sufficient additional proof of claim sufficient to prove entitlement to DSC claim in spite of slight prejudice to the government? (Yes/No)
: No

Inadequacy of or lack of notice plus resulting prejudice to the government render entitlement to DSC claim unlikely. However, I will continue our analysis.

[14] Who expressly assumed obligation for or risk of the condition in the contract?
a. The government
b. The contractor
c. Both the government and the contractor
d. Neither the government nor the contractor
: d

[15] Is the condition in question directly related to the physical conditions at the work site? (Yes/No)
: Yes

[16] Is the condition a static physical part of the work site? (Yes/No)
: Yes

[17] Did the condition occur before or after the contract award?
: Before

[18] Were there affirmatively expressed statements in the contract concerning the relevant subsurface or latent conditions at the site? (Yes/No)
: -

[19] Would the affirmatively expressed contract statements on the general conditions at the site lead a reasonable contractor to believe that the condition could be expected? (Yes/No)
: -

[20] Were indications of subsurface or latent conditions inferred from the contract as a whole? (Yes/No)
: Yes

[21] Was this inference justified based on the information in the contract? (Yes/No)
: Yes

There were indications regarding the condition in the contract, therefore, a Type I condition is the most probable avenue of recovery.

[22] Were there latent deficiencies in the contract indications? (Yes/No)
: Yes

[23] Was the physical data made a part of the contract? (Yes/No)
: Yes

[24] Would this physical data have revealed or resolved the apparent deficiencies in the contract indications for a reasonable contractor? (Yes/No)
: Yes

Thinking.......

**** Entitlement will probably not be allowed under DSC claim. ****

I believe that entitlement will probably not be allowed because I concluded that the contractor did fail to-heed contract indications.

The following rule proved to be true causing me to draw the above conclusion:

If
'contract indications did not contain latent deficiencies'
is true in answers
or ('contract information did contain physical data'
is true in answers

and 'contract information did reveal deficiencies/conditions'
is true in answers)
then the contractor did fail to-heed contract indications.

The following statements contained by the rule
are true:

[1] contract information did contain physical data (in answers)

[2] contract information did reveal deficiencies/conditions (in answers)

********************* END OF ANALYSIS *********************
(You should receive my bill in 7-10 days.)
(PROMPT PAYMENT IS APPRECIATED)

Do you wish to see the results from this session? (Yes/No)
: Yes

Which of the following results do you wish
to see?

a. The answers given to the questions
b. The unknown information
c. The assumptions which have been made
d. The conclusions which have been reached
e. All of the above
f. None of the above

: e

These are the results of this session:

The answers:
[ANSWERS Database]
CONDITION did occur before CONTRACT AWARD.
CONTRACTOR did have POTENTIAL-TO-KNOW CONDITION SOONER.
PASSAGE-OF-TIME did obscure EVIDENCE.
GOVERNMENT did not expressly ASSUME OBLIGATION-RISK.
CONTRACTOR did not expressly ASSUME OBLIGATION-RISK.
CONTRACT did lead ONE-TO INFER CONDITIONS.
CONTRACT INFORMATION did justify INFERENCE.
CONTRACT INDICATIONS did contain LATENT DEFICIENCIES.
CONTRACT INFORMATION did contain PHYSICAL DATA.
CONTRACT INFORMATION did reveal DEFICIENCIES/CONDITIONS.
ADDITIONAL-PROOF does not exist TO-PROVE-ENTITLEMENT.
<JULY 26, 1968> is a date-of-dsc-assertion.
<JULY 26, 1968> is a date-of-written-notice.
NOTICE was received-by CONTRACTING OFFICER.
CONDITION is static PHYSICAL PART OF-WORK-SITE.

DSC was asserted.
FINAL-PAYMENT was not made.
FORM-OF-NOTICE was written.
NOTICE was not given before REMEDIAL WORK PERFORMED.
CONDITION is directly-related-to-physical-conditions.

The unknowns:
[UNKNOWNS Database]
IF-IMPROPER-NOTICE did cause ADDITIONAL COST.
IF-ANYTHING-ELSE did cause PREJUDICE.
IF-CONTRACT did contain STATEMENTS CONCERNING CONDITION.
IF-CONTRACT did contain GENERAL CONDITIONS IMPLYING CONDITION.
IF-DEFENSE-AGAINST CLAIM was made IMPOSSIBLE.

The assumptions:
[ASSUMPTIONS Database]
IMPROPER-NOTICE did not cause ADDITIONAL COST.
NOTHING-ELSE did cause PREJUDICE.
CONTRACT did not contain STATEMENTS CONCERNING CONDITION.
CONTRACT did not contain GENERAL CONDITIONS IMPLYING CONDITION.
DEFENSE-AGAINST CLAIM was not made IMPOSSIBLE.

The conclusions:
[REPORTS Database]
CONTRACTOR did comply with WRITTEN-NOTICE-REQUIREMENT.
REASON_FOR NO ENTITLEMENT does exist.
GOVERNMENT did receive NOTICE-OF-CONDITION.
PASSAGE-OF-TIME did cause PREJUDICE.
GOVERNMENT did suffer PREJUDICE.
CONTRACT did not assign OBLIGATION-RISK.
CONTRACT did contain INDICATIONS CONCERNING CONDITION.
CONTRACTOR did fail TO-HEED CONTRACT INDICATIONS.
PROMPTNESS_OF_NOTICE is an in_doubt.
NOTICE was given-to RESPONSIBLE RECEIVER.

Which of the other results do you want to see?

a. The answers given to the questions
b. The unknown information
c. The assumptions which have been made
d. The conclusions which have been reached
e. All of the above
f. None of the above

: f

Do you wish to store this session? (Yes/No)
: Yes

Please enter your name or initials. (Please place a '-'

between initials and/or name.)
: TAK

Session has been saved as requested.

75-1 BCA

[¶ 10, 997] J. J. Welcome Construction Company, Inc.

ASBCA No. 19653. December 6, 1974.

Contract No. HSM 79-74-10.

Changed Conditions--Notice of Changed Conditions Existing--Substantial Compliance with Notice Requirements.

A contractor was entitled to an equitable adjustment under a HEW contract for the construction of a sewage system and lagoon because of a changed condition in the work-site created by the presence of a bottomless muck-like soil condition which prevented the use of heavy equipment intended for use in clearing and grading the site. The contractor was issued a change order for the importation of 26,000 yards of rock in order to stabilize this subsurface condition which was not disclosed by either his site investigation or examination of a test hole sample of the typical soil profile of the area. The government contended that the contractor's failure to give it notice that the unanticipated condition would result in additional clearing costs should bar his recovery. However, the board held that the Differing Site clause did not require him to notify the government of anticipated additional costs, but only of the existence of a changed condition. Therefore, since the contractor had already been issued a change order for the importation of rock fill to stabilize the changed condition, a decision in which the government's engineers had concurred, the government was

held to have had notice of the condition. Moreover, since it was aware of the defective condition of the soil and failed to change the location of the lagoon, the contractor had no choice but to incur the added clearing expenses necessary to provide a site which would support the lagoon dike, and constitute a quality performance which was in keeping with his considerable experience.

TM
[Rosie Version 2.3 30-NOV-83 16:23:51]

The name of this session shall be: JJWELCO

You may perform any one of the following analyses:

DSC Assertion Final Payment Notice Form
Notice Promptness Responsible Receiver Government Prejudiced

Contract Obligation Excluded Conditions After-bid Conditions

Express-implied Conditions Contract Indications
Latent Deficiencies Standard Conditions
Site Inspection

Superior Knowledge-I Superior Knowledge-II
Reliance I Reliance II
Material Difference-I Material Difference-II
Exculpatory Langauge-I Exculpatory Language-II

Complete Analysis No Module

I will begin the analysis as soon as you have selected any ONE of the options listed. Please enter the selected option EXACTLY as it appears in the list followed by a '.'.
: Complete Analysis.

SESSION IN PROGRESS

[1] Has a differing site condition been asserted by one of your contractors? (Yes/No)
: Yes

[2] On what date did the DSC-ASSERTION occur (month day, year)
: November 8, 1973

[3] Has final payment been made? (Yes/No)
: No

[4] What form of notice was given to the government by the contractor? (Select one of the following.)

a. Written notice was given.
b. Oral notice was given.
c. Other events surrounded notice.

: b

[5] On what date did the ORAL-NOTICE occur (month day, year)
: July 1973

[6] When was notice given in relation to the progression of work?
a. Notice given before contractor disturbed the condition.
b. Notice given before contractor performed remedial work.
c. Both a and b.
: c

[7] Is there evidence that the contractor should have been aware of the differing site condition sooner? (Yes/No)
: No

[8] To whom was notice of the differing site condition given?
a. The contracting officer
b. An authorized representative of the contracting officer
c. Any other employee of the government
d. None of the above
: -

Receipt of notice by responsible government employee is in doubt.

[9] Is the government able to show that evidence against the claim was obscurred due to the passage of time? (Yes/No)
: No

[10] Is the government able to show that proper notice would have resulted in cheaper resolution of the condition by CO? (Yes/No)
: No

[11] Can the government show that it suffered any other prejudice through inadequacy of or lack of notice? (Yes/No)
: No

Government can't prove that it suffered prejudice.

[12] Who expressly assumed obligation for or risk of the

condition in the contract?

a. The government

b. The contractor

c. Both the government and the contractor

d. Neither the government nor the contractor

: d

[13] Is the condition in question directly related to the physical conditions at the work site? (Yes/No)

: Yes

[14] Is the condition a static physical part of the work site? (Yes/No)

: Yes

[15] Did the condition occur before or after the contract award?

: Before

[16] Were there affirmatively expressed statements in the contract concerning the relevant subsurface or latent conditions at the site? (Yes/No)

: No

[17] Would the affirmatively expressed contract statements on the general conditions at the site lead a reasonable contractor to believe that the condition could be expected? (Yes/No)

: No

[18] Were indications of subsurface or latent conditions inferred from the contract as a whole? (Yes/No)

: No

There were no indications regarding the condition in the contract, therefore, a Type II condition is the most probable avenue of recovery.

[19] Was the physical data made a part of the contract? (Yes/No)

: -

[20] By what means can the contractor establish expected conditions?
a. Customs of the trade
b. General knowledge in the industry
c. Manufacturers instructions or recommendations
d. None of the above
: b

[21] Does the actual condition differ from the standard? (Yes/No)
: Yes

[22] Did the GOVERNMENT know about the ACTUAL-CONDITIONS? (Yes/No)
: -

[23] Was the contractor in possession of unilateral superior knowledge of the actual conditions? (Yes/No)
: No

[24] Would simple inquiries by the contractor have revealed the condition to be contrary to the STANDARD-OF-EXPECTATION? (Yes/No)
: No

[25] Did any of the following acts of the government hinder the site inspection?

a. Access to site was denied
b. Inadequate time was allowed
c. No act of the government hindered inspection
: c

[26] Did the contractor conduct a site inspection? (Yes/No)
: Yes

[27] Was the inspection conducted by the contractor a reasonable inspection, i.e., one comparable to that expected from a reasonable, prudent contractor experienced in that particular field of work? (Yes/No)
: -

[28] Would the actual condition have been discernable
by a layman contractor performing a reasonable
inspection? (Yes/No)
: No

[29] Did the contractor reasonably rely on
EXPECTATIONS-THAT-CONDITION-WOULD-MEET-STANDARD
when preparing his bid causing him to suffer prejudice
through this reliance? (Yes/No)
: Yes

[30] Was the difference between actual and EXPECTED
conditions a material difference? (Yes/No)
: Yes

Relief is probably available through the Type II
DSC clause because of unanticipated or unusual
conditions or materials.

[31] Is there a clause in the contract placing the risk
of conditions not revealed by the contractor's site
inspection? (Yes/No)
: No

Since the contract does not contain an exculpatory
clause entitlement will probably be allowed.

Consult your local CO for further details.

Thinking.......

********************* END OF ANALYSIS *********************
(You should receive my bill in 7-10 days.)
(PROMPT PAYMENT IS APPRECIATED)

Do you wish to see the results from this session? (Yes/No)
: Yes

Which of the following results do you wish
to see?

a. The answers given to the questions
b. The unknown information
c. The assumptions which have been made
d. The conclusions which have been reached
e. All of the above
f. None of the above

: e

These are the results of this session:

The answers:
[ANSWERS Database]
CONDITION did occur before CONTRACT AWARD.
CONDITION does differ from STANDARD-OF-EXPECTATION.
CONTRACTOR did not have POTENTIAL-TO-KNOW CONDITION SOONER.
REASONABLE INSPECTION did not have POTENTIAL-TO-REVEAL CONDITION.
PASSAGE-OF-TIME did not obscure EVIDENCE.
IMPROPER-NOTICE did not cause ADDITIONAL COST.
NOTHING-ELSE did cause PREJUDICE.
GOVERNMENT did not expressly ASSUME OBLIGATION-RISK.
CONTRACTOR did not expressly ASSUME OBLIGATION-RISK.
CONTRACT did not contain STATEMENTS CONCERNING CONDITION.
CONTRACT did not contain GENERAL CONDITIONS IMPLYING CONDITION.
CONTRACT did not lead ONE-TO INFER CONDITIONS.
SIMPLE INQUIRY did not give POTENTIAL TO-KNOW CONTRARY CONDITIONS.
GOVERNMENT did not hinder INSPECTION.
CONTRACTOR did conduct SITE-INSPECTION.
CONTRACTOR did suffer PREJUDICE through RELIANCE.
CONTRACT does not contain EXCULPATORY-CLAUSE.
<NOVEMBER 8, 1973> is a date-of-dsc-assertion.
<JULY 1973> is a date-of-oral-notice.
CONTRACTOR was not aware-of ACTUAL-CONDITIONS.
CONDITION is static PHYSICAL PART OF-WORK-SITE.
DSC was asserted.
FINAL-PAYMENT was not made.
FORM-OF-NOTICE was oral.
NOTICE was given before CONTRACTOR DISTURBED CONDITION.
NOTICE was given before REMEDIAL WORK PERFORMED.
DIFFERENCE was material.
CONDITION is directly-related-to-physical-conditions.
EXPECTED CONDITIONS is establishable from GENERAL-KNOWLEDGE-IN-INDUSTRY
.

The unknowns:
[UNKNOWNS Database]
IF-CONTRACT INFORMATION did contain PHYSICAL DATA.
TO-WHOM was notice GIVEN.
IF-GOVERNMENT was aware-of ACTUAL-CONDITIONS.
IF-SITE-INSPECTION was reasonable INSPECTION.

The assumptions:

[ASSUMPTIONS Database]
CONTRACT INFORMATION did not contain PHYSICAL DATA.
NOTICE was not received-by GOVERNMENT.
GOVERNMENT was not aware-of ACTUAL-CONDITIONS.
SITE-INSPECTION was reasonable INSPECTION.

The conclusions:
[REPORTS Database]
CONTRACTOR did constructively COMPLY with NOTICE-REQUIREMENT.
GOVERNMENT did not suffer PREJUDICE.
CONTRACT did not assign OBLIGATION-RISK.
CONTRACT did not contain INDICATIONS CONCERNING CONDITION.
REASONABLE INSPECTION did not reveal CONDITION-**.
RECEIPT-OF-NOTICE is an in_doubt.
CONTRACTOR is able-to ESTABLISH STANDARD-OF-EXPECTATION.
NOTICE was prompt.
REASONABLE INSPECTION is not required-for-entitlement.
ENTITLEMENT is still-probable.

Which of the other results do you want to see?

a. The answers given to the questions
b. The unknown information
c. The assumptions which have been made
d. The conclusions which have been reached
e. All of the above
f. None of the above

: f

Do you wish to store this session? (Yes/No)
: Yes

Please enter your name or initials. (Please place a '-'
between initials and/or name.)
: TAK

Session has been saved as requested.

TM
[Rosie Version 2.3 1-DEC-83 20:35:49]

The name of this session shall be: JJWELCO

Author of previous session: TAK

Date of previous session: 30-NOV-83

You may perform any one of the following analyses:

DSC Assertion Final Payment Notice Form
Notice Promptness Responsible Receiver Government Prejudiced

Contract Obligation Excluded Conditions After-bid Conditions

Express-implied Conditions Contract Indications
Latent Deficiencies Standard Conditions
Site Inspection

Superior Knowledge-I Superior Knowledge-II
Reliance I Reliance II
Material Difference-I Material Difference-II
Exculpatory Langauge-I Exculpatory Language-II

Complete Analysis No Module

I will begin the analysis as soon as you have selected any ONE
of the options listed. Please enter the selected option
EXACTLY as it appears in the list followed by a '.'.
: Complete Analysis.

SESSION IN PROGRESS

The differing site condition has been asserted.
The date of assertion of the differing site
condition is <NOVEMBER 8, 1973>.

The final payment was not made.

The contractor gave oral notice, therefore,the
contractor constructively complied with the

notice requirement.
The oral notice was given on <JULY 1973>.

The notice was given before disturbing the condition and before performing the work.
Therefore, notice was prompt.

The contractor should not have been aware of the condition sooner.

Do you know to whom notice of the DSC was given yet? (Yes/No)
: No

Neither the government nor the contractor have assumed obligation or risk for the condition.

The condition is directly related to the physical conditions at the site.

The condition is a static physical part of the work site.

The condition did occur before the contract was awarded

There were no indications regarding the condition in the contract, therefore, a Type II condition is the most probable avenue of recovery.

Do you know yet whether or not the physical data was made a part of the contract? (Yes/No)
: No

The contractor can establish expected conditions from general knowledge in the industry.

The actual condition does differ from the standard of expectation.

Do you know yet whether or not the GOVERNMENT knew

about or anticipated the ACTUAL-CONDITIONS? (Yes/No)
: No

The contractor was not aware of the ACTUAL-CONDITIONS.

Simple inquiries by the contractor would not
have revealed conditions to be contrary to those
indicated.

The government did not hinder the site inspection
in any way.

The contractor did conduct a site inspection.

Do you know yet whether or not the contractor
conducted a reasonable site inspection? (Yes/No)
: No

Actual conditions would not have been discernable
from a reasonable inspection, therefore, reasonable
site inspection is not required for entitlement.

The contractor did suffer prejudice through his
reliance on the EXPECTATIONS-THAT-CONDITION-WOULD-MEET-STANDARD.

The difference between actual and EXPECTED
conditions is material.

Relief is probably available through the Type II
DSC clause because of unanticipated or : iusual
conditions or materials.

The contract does not contain a site inspection
exculpatory clause.

Therefore, entitlement will probably be allowed.

Consult your local CO for further details.

Thinking.......

********************** END OF ANALYSIS **********************
(You should receive my bill in 7-10 days.)
(PROMPT PAYMENT IS APPRECIATED)

Do you wish to see the results from this session? (Yes/No)
: Yes

Which of the following results do you wish
to see?

a. The answers given to the questions
b. The unknown information
c. The assumptions which have been made
d. The conclusions which have been reached
e. All of the above
f. None of the above

: e

These are the results of this session:

The answers:
[ANSWERS Database]
CONDITION did occur before CONTRACT AWARD.
CONDITION does differ from STANDARD-OF-EXPECTATION.
CONTRACTOR did not have POTENTIAL-TO-KNOW CONDITION SOONER.
REASONABLE INSPECTION did not have POTENTIAL-TO-REVEAL CONDITION.
PASSAGE-OF-TIME did not obscure EVIDENCE.
IMPROPER-NOTICE did not cause ADDITIONAL COST.
NOTHING-ELSE did cause PREJUDICE.
GOVERNMENT did not expressly ASSUME OBLIGATION-RISK.
CONTRACTOR did not expressly ASSUME OBLIGATION-RISK.
CONTRACT did not contain STATEMENTS CONCERNING CONDITION.
CONTRACT did not contain GENERAL CONDITIONS IMPLYING CONDITION.
CONTRACT did not lead ONE-TO INFER CONDITIONS.
SIMPLE INQUIRY did not give POTENTIAL TO-KNOW CONTRARY CONDITIONS.
GOVERNMENT did not hinder INSPECTION.
CONTRACTOR did conduct SITE-INSPECTION.
CONTRACTOR did suffer PREJUDICE through RELIANCE.
CONTRACT does not contain EXCULPATORY-CLAUSE.
<NOVEMBER 8, 1973> is a date-of-dsc-assertion.
<JULY 1973> is a date-of-oral-notice.
CONTRACTOR was not aware-of ACTUAL-CONDITIONS.
CONDITION is static PHYSICAL PART OF-WORK-SITE.
DSC was asserted.
FINAL-PAYMENT was not made.
FORM-OF-NOTICE was oral.
NOTICE was given before CONTRACTOR DISTURBED CONDITION.

NOTICE was given before REMEDIAL WORK PERFORMED.
DIFFERENCE was material.
CONDITION is directly-related-to-physical-conditions.
EXPECTED CONDITIONS is establishable from GENERAL-KNOWLEDGE-IN-INDUSTRY
.

The unknowns:
[UNKNOWNS Database]
IF-CONTRACT INFORMATION did contain PHYSICAL DATA.
TO-WHOM was notice GIVEN.
IF-GOVERNMENT was aware-of ACTUAL-CONDITIONS.
IF-SITE-INSPECTION was reasonable INSPECTION.

The assumptions:
[ASSUMPTIONS Database]
CONTRACT INFORMATION did not contain PHYSICAL DATA.
GOVERNMENT was not aware-of ACTUAL-CONDITIONS.
SITE-INSPECTION was reasonable INSPECTION.
NOTICE was not received by GOVERNMENT.

The conclusions:
[REPORTS Database]
CONTRACTOR did comply-fully with NOTICE-REQUIREMENT.
CONTRACTOR did constructively COMPLY with NOTICE-REQUIREMENTS.
CONTRACT did not assign OBLIGATION-RISK.
CONTRACT did not contain INDICATIONS CONCERNING CONDITION.
CONTRACTOR is able-to ESTABLISH STANDARD-OF-EXPECTATION.
NOTICE was prompt.
RECEIPT-OF-NOTICE is in_doubt.
REASONABLE INSPECTION is not required-for-entitlement.
ENTITLEMENT is still-probable.

Which of the other results do you want to see?

a. The answers given to the questions
b. The unknown information
c. The assumptions which have been made
d. The conclusions which have been reached
e. All of the above
f. None of the above

: f

Do you wish to store this session? (Yes/No)
: No

Session NOT saved.

65-1 BCA

[¶ 4658] LAYNE TEXAS CO.

IBCA No. 362. January 29, 1965.

Contract No. 14-06-400-1745.

Specifications--Changed Conditions--Subsurface Boulders.--Boulders of up to 12" in diameter, encountered in drilling test holes and water supply wells on a basin project, were not conditions materially different from those indicated in a specification stating that "drilling will probably be through clay, sand, and gravel formations" in alluvial and lake deposits. The contractor anticipated some boulders and the specification was silent as to the percentage of subsurface boulders that might be encountered.

Changed Conditions--Unanticipated Condition--Subsurface Boulders.--A contractor hired by the Government to drill test holes and water supply wells on a basin project was entitled to an equitable adjustment under a changed conditions clause. The contractor encountered an unanticipated condition to the extent that more than 20% of the soil consisted of subsurface boulders. This was the normal ratio to be expected according to the contractor's site inspection, performance of an earlier drilling contract in the same area, and general history of the terrain.

TM
[Rosie Version 2.3 5-DEC-83 01:41:31]

The name of this session shall be: LAYNE

You may perform any one of the following analyses:

DSC Assertion Final Payment Notice Form
Notice Promptness Responsible Receiver Government Prejudiced

Contract Obligation Excluded Conditions After-bid Conditions

Express-implied Conditions Contract Indications
Latent Deficiencies Standard Conditions
Site Inspection

Superior Knowledge-I Superior Knowledge-II
Reliance I . Reliance II
Material Difference-I Material Difference-II
Exculpatory Langauge-I Exculpatory Language-II

Complete Analysis No Module

I will begin the analysis as soon as you have selected any ONE of the options listed. Please enter the selected option EXACTLY as it appears in the list followed by a '.'.
: Complete Analysis.

SESSION IN PROGRESS

[1] Has a differing site condition been asserted by one of your contractors? (Yes/No)
: Yes

[2] On what date did the DSC-ASSERTION occur? (month day, year)
: -

[3] Has final payment been made? (Yes/No)
: No

[4] What form of notice was given to the government by the contractor? (Select one of the following.)

a. Written notice was given.
b. Oral notice was given.
c. Other events surrounded notice.

: b

[5] On what date did the ORAL-NOTICE occur? (month day, year)
: -

[6] When was notice given in relation to the progression of work?
a. Notice given before contractor disturbed the condition.
b. Notice given before contractor performed remedial work.
c. Both a and b.
d. None of the above.
: c

[7] Is there evidence that the contractor should have been aware of the differing site condition sooner? (Yes/No)
: No

[8] To whom was notice of the differing site condition given?
a. The contracting officer
b. An authorized representative of the contracting officer
c. Any other employee of the government
d. None of the above
: b

[9] Who expressly assumed obligation for or risk of the condition in the contract?
a. The government
b. The contractor
c. Both the government and the contractor
d. Neither the government nor the contractor
: d

[10] Is the condition in question directly related to the physical conditions at the work site? (Yes/No)
: Yes

[11] Is the condition a static physical part of the work site? (Yes/No)
: Yes

[12] Did the condition occur before or after the contract award?
: Before

[13] Were there affirmatively expressed statements in the contract concerning the relevant subsurface or latent conditions at the site? (Yes/No)
: No

[14] Would the affirmatively expressed contract statements on the general conditions at the site lead a reasonable contractor to believe that the condition could be expected? (Yes/No)
: No

[15] Were indications of subsurface or latent conditions inferred from the contract as a whole? (Yes/No)
: No

There were no indications regarding the condition in the contract, therefore, a Type II condition is the most probable avenue of recovery.

[16] Was the physical data made a part of the contract? (Yes/No)
: Yes

[17] Would this physical data have indicated the nature of existing conditions to a reasonable contractor? (Yes/No)
: No

[18] By what means can the contractor establish expected conditions?
a. Customs of the trade
b. General knowledge in the industry
c. Manufacturers instructions or recommendations
d. None of the above
: b

[19] Does the actual condition differ from the standard? (Yes/No)
: Yes

[20] Did the GOVERNMENT know about the ACTUAL-CONDITIONS? (Yes/No)
: No

[21] Was the contractor in possession of unilateral superior knowledge of the actual conditions? (Yes/No)
: No

[22] Would simple inquiries by the contractor have revealed the condition to be contrary to the STANDARD-OF-EXPECTATION? (Yes/No)
: No

[23] Did any of the following acts of the government hinder the site inspection?

a. Access to site was denied
b. Inadequate time was allowed
c. No act of the government hindered inspection
: c

[24] Did the contractor conduct a site inspection? (Yes/No)
: Yes

[25] Was the inspection conducted by the contractor a reasonable inspection, i.e., one comparable to that expected from a reasonable, prudent contractor experienced in that particular field of work? (Yes/No)
: -

[26] Would the actual condition have been discernable by a layman contractor performing a reasonable inspection? (Yes/No)
: No

[27] Did the contractor reasonably rely on EXPECTATIONS-THAT-CONDITION-WOULD-MEET-STANDARD when preparing his bid causing him to suffer prejudice through this reliance? (Yes/No)
: Yes

[28] Was the difference between actual and EXPECTED conditions a material difference? (Yes/No)
: Yes

Relief is probably available through the Type II DSC clause because of unanticipated or unusual conditions or materials.

[29] Is there a clause in the contract placing the risk of conditions not revealed by 'the contractor's site inspection? (Yes/No)
: No

Since the contract does not contain an exculpatory clause entitlement will probably be allowed.

Consult your local CO for further details.

Thinking.......

********************* END OF ANALYSIS *********************
(You should receive my bill in 7-10 days.)
(PROMPT PAYMENT IS APPRECIATED)

Do you wish to see the results from this session? (Yes/No)
: Yes

Which of the following results do you wish to see?

a. The answers given to the questions
b. The unknown information
c. The assumptions which have been made
d. The conclusions which have been reached
e. All of the above
f. None of the above

: e

These are the results of this session:

The answers:
[ANSWERS Database]
CONDITION did occur before CONTRACT AWARD.
CONDITION does differ from STANDARD-OF-EXPECTATION.
CONTRACTOR did not have POTENTIAL-TO-KNOW CONDITION SOONER.
REASONABLE INSPECTION did not have POTENTIAL-TO-REVEAL CONDITION.
GOVERNMENT did not expressly ASSUME OBLIGATION-RISK.
CONTRACTOR did not expressly ASSUME OBLIGATION-RISK.
CONTRACT did not contain STATEMENTS CONCERNING CONDITION.

CONTRACT did not contain GENERAL CONDITIONS IMPLYING CONDITION.
CONTRACT INFORMATION did contain PHYSICAL DATA.
CONTRACT did not lead ONE-TO INFER CONDITIONS.
CONTRACT INFORMATION did not reveal DEFICIENCIES/CONDITIONS.
SIMPLE INQUIRY did not give POTENTIAL TO-KNOW CONTRARY CONDITIONS.
GOVERNMENT did not hinder INSPECTION.
CONTRACTOR did conduct SITE-INSPECTION.
CONTRACTOR did suffer PREJUDICE through RELIANCE.
CONTRACT does not contain EXCULPATORY-CLAUSE.
<UNKNOWN> is a date-of-dsc-assertion.
<UNKNOWN> is a date-of-oral-notice.
NOTICE was received-by AUTHORIZED REPRESENTATIVE.
GOVERNMENT was not aware-of ACTUAL-CONDITIONS.
CONTRACTOR was not aware-of ACTUAL-CONDITIONS.
!" CONDITION is static PHYSICAL PART OF-WORK-SITE.
DSC was asserted.
FINAL-PAYMENT was not made.
FORM-OF-NOTICE was oral.
NOTICE was given before CONTRACTOR DISTURBED CONDITION.
NOTICE was given before REMEDIAL WORK PERFORMED.
DIFFERENCE was material.
CONDITION is directly-related-to-physical-conditions.
EXPECTED CONDITIONS is establishable from GENERAL-KNOWLEDGE-IN-INDUSTRY
.

The unknowns:
[UNKNOWNS Database]
IF-SITE-INSPECTION was reasonable INSPECTION.

The assumptions:
[ASSUMPTIONS Database]
SITE-INSPECTION was reasonable INSPECTION.

The conclusions:
[REPORTS Database]
CONTRACTOR did comply-fully with NOTICE-REQUIREMENT.
CONTRACTOR did constructively COMPLY with NOTICE-REQUIREMENT.
CONTRACT did not assign OBLIGATION-RISK.
CONTRACT did not contain INDICATIONS CONCERNING CONDITION.
REASONABLE INSPECTION did not reveal CONDITION.
NOTICE was given-to RESPONSIBLE RECEIVER.
CONTRACTOR is not required-to INSPECT OTHER DATA.
CONTRACTOR is able-to ESTABLISH STANDARD-OF-EXPECTATION.
NOTICE was prompt.
REASONABLE INSPECTION is not required-for-entitlement.
ENTITLEMENT is still-probable.

Which of the other results do you want to see?

a. The answers given to the questions
b. The unknown information
c. The assumptions which have been made
d. The conclusions which have been reached
e. All of the above
f. None of the above

: f

Do you wish to store this session? (Yes/No)
: Yes

Please enter your name or initials. (Please place a '-' between initials and/or name.)
: TaK

Session has been saved as requested.

TM
[Rosie Version 2.3 6-DEC-83 04:44:57]

The name of this session shall be: LAYNE

Author of previous session: TAK

Date of previous session: 5-DEC-83

You may perform any one of the following analyses:

DSC Assertion Final Payment Notice Form
Notice Promptness Responsible Receiver Government Prejudiced

Contract Obligation Excluded Conditions After-bid Conditions

Express-implied Conditions Contract Indications
Latent Deficiencies Standard Conditions
Site Inspection

Superior Knowledge-I Superior Knowledge-II
Reliance I Reliance II
Material Difference-I Material Difference-II
Exculpatory Langauge-I Exculpatory Language-II

Complete Analysis No Module

I will begin the analysis as soon as you have selected any ONE
of the options listed. Please enter the selected option
EXACTLY as it appears in the list followed by a '.'.
: Complete Analysis.

SESSION IN PROGRESS

The differing site condition has been asserted.
The date of assertion of the differing site
condition is <UNKNOWN>.

The final payment was not made.

The contractor gave oral notice, therefore,the
contractor constructively complied with the

notice requirement.
The oral notice was given on <UNKNOWN>.

The notice was given before disturbing the condition and before performing the work.
Therefore, notice was prompt.

The contractor should not have been aware of the condition sooner.

Notice of DSC was given to an authorized representative of the CO.

Neither the government nor the contractor have assumed obligation or risk for the condition.

The condition is directly related to the physical conditions at the site.

The condition is a static physical part of the work site.

The condition did occur before the contract was awarded.

There were no indications regarding the condition in the contract, therefore, a Type II condition is the most probable avenue of recovery.

The physical data was made part of the contract.

The physical data contained in the contract would not have indicated the nature of the existing condition.

The contractor can establish expected conditions from general knowledge in the industry.

The actual condition does differ from the standard of expectation.

The government was not aware of the ACTUAL-CONDITIONS.

The contractor was not aware of the ACTUAL-CONDITIONS.

Simple inquiries by the contractor would not have revealed conditions to be contrary to those indicated.

The government did not hinder the site inspection in any way.

The contractor did conduct a site inspection.

Do you know yet whether or not the contractor conducted a reasonable site inspection? (Yes/No)
: No

Actual conditions would not have been discernable from a reasonable inspection, therefore, reasonable site inspection is not required for entitlement.

The contractor did suffer prejudice through his reliance on the EXPECTATIONS-THAT-CONDITION-WOULD-MEET-STANDARD.

The difference between actual and EXPECTED conditions is material.

Relief is probably available through the Type II DSC clause because of unanticipated or unusual conditions or materials.

The contract does not contain a site inspection exculpatory clause.
Therefore, entitlement will probably be allowed.
Consult your local CO for further details.

Thinking.......

********************* END OF ANALYSIS *********************
(You should receive my bill in 7-10 days.)
(PROMPT PAYMENT IS APPRECIATED)

Do you wish to see the results from this session? (Yes/No)
: Yes

Which of the following results do you wish
to see?

a. The answers given to the questions
b. The unknown information
c. The assumptions which have been made
d. The conclusions which have been reached
e. All of the above
f. None of the above

: e

These are the results of this session:

The answers:
[ANSWERS Database]
CONDITION did occur before CONTRACT AWARD.
CONDITION does differ from STANDARD-OF-EXPECTATION.
CONTRACTOR did not have POTENTIAL-TO-KNOW CONDITION SOONER.
REASONABLE INSPECTION did not have POTENTIAL-TO-REVEAL CONDITION.
GOVERNMENT did not expressly ASSUME OBLIGATION-RISK.
CONTRACTOR did not expressly ASSUME OBLIGATION-RISK.
CONTRACT did not contain STATEMENTS CONCERNING CONDITION.
CONTRACT did not contain GENERAL CONDITIONS IMPLYING CONDITION.
CONTRACT INFORMATION did contain PHYSICAL DATA.
CONTRACT did not lead ONE-TO INFER CONDITIONS.
CONTRACT INFORMATION did not reveal DEFICIENCIES/CONDITIONS.
SIMPLE INQUIRY did not give POTENTIAL TO-KNOW CONTRARY CONDITIONS.
GOVERNMENT did not hinder INSPECTION.
CONTRACTOR did conduct SITE-INSPECTION.
CONTRACTOR did suffer PREJUDICE through RELIANCE.
CONTRACT does not contain EXCULPATORY-CLAUSE.
<UNKNOWN> is a date-of-dsc-assertion.
<UNKNOWN> is a date-of-oral-notice.
NOTICE was received-by AUTHORIZED REPRESENTATIVE.
GOVERNMENT was not aware-of ACTUAL-CONDITIONS.
CONTRACTOR was not aware-of ACTUAL-CONDITIONS.
CONDITION is static PHYSICAL PART OF-WORK-SITE.
DSC was asserted.
FINAL-PAYMENT was not made.
FORM-OF-NOTICE was oral.
NOTICE was given before CONTRACTOR DISTURBED CONDITION.
NOTICE was given before REMEDIAL WORK PERFORMED.
DIFFERENCE was material.
CONDITION is directly-related-to-physical-conditions.

EXPECTED CONDITIONS is establishable from GENERAL-KNOWLEDGE-IN-INDUSTRY

.

The unknowns:
[UNKNOWNS Database]
IF-SITE-INSPECTION was reasonable INSPECTION.

The assumptions:
[ASSUMPTIONS Database]
SITE-INSPECTION was reasonable INSPECTION.

The conclusions:
[REPORTS Database]
CONTRACTOR did comply-fully with NOTICE-REQUIREMENT.
CONTRACTOR did constructively COMPLY with NOTICE-REQUIREMENTS.
GOVERNMENT did receive NOTICE-OF-CONDITION.
CONTRACT did not assign OBLIGATION-RISK.
CONTRACT did not contain INDICATIONS CONCERNING CONDITION.
CONTRACTOR is not required-to INSPECT OTHER DATA.
CONTRACTOR is able-to ESTABLISH STANDARD-OF-EXPECTATION.
NOTICE was prompt.
NOTICE was given to RESPONSIBLE RECEIVER.
REASONABLE INSPECTION is not required-for-entitlement.
ENTITLEMENT is still-probable.

Which of the other results do you want to see?

a. The answers given to the questions
b. The unknown information
c. The assumptions which have been made
d. The conclusions which have been reached
e. All of the above
f. None of the above

: f

Do you wish to store this session? (Yes/No)
: No

Session NOT saved.

TM
[Rosie Version 2.3 5-DEC-83 01:55:09]

The name of this session shall be: LAYNE2

You may perform any one of the following analyses:

DSC Assertion   Final Payment   Notice Form
Notice Promptness   Responsible Receiver   Government Prejudiced

Contract Obligation   Excluded Conditions   After-bid Conditions

Express-implied Conditions   Contract Indications
Latent Deficiencies   Standard Conditions
Site Inspection

Superior Knowledge-I   Superior Knowledge-II
Reliance I   Reliance II
Material Difference-I   Material Difference-II
Exculpatory Langauge-I   Exculpatory Language-II

Complete Analysis   No Module

I will begin the analysis as soon as you have selected any ONE
of the options listed. Please enter the selected option
EXACTLY as it appears in the list followed by a '.'.
: Complete Analysis.

SESSION IN PROGRESS

[1] Has a differing site condition been asserted by one of
your contractors? (Yes/No)
: Yes

[2] On what date did the DSC-ASSERTION occur? (month day, year)
: -

[3] Has final payment been made? (Yes/No)
: No

[4] What form of notice was given to the government by
the contractor? (Select one of the following.)
a. Written notice was given.
b. Oral notice was given.
c. Other events surrounded notice.

: b

[5] On what date did the ORAL-NOTICE occur? (month day, year)
: -

[6] When was notice given in relation to the progression of work?
a. Notice given before contractor disturbed the condition.
b. Notice given before contractor performed remedial work.
c. Both a and b.
d. None of the above.
: c

[7] Is there evidence that the contractor should have been aware of the differing site condition sooner? (Yes/No)
: No

[8] To whom was notice of the differing site condition given?
a. The contracting officer
b. An authorized representative of the contracting officer
c. Any other employee of the government
d. None of the above
: b

[9] Who expressly assumed obligation for or risk of the condition in the contract?
a. The government
b. The contractor
c. Both the government and the contractor
d. Neither the government nor the contractor
: d

[10] Is the condition in question directly related to the physical conditions at the work site? (Yes/No)
: Yes

[11] Is the condition a static physical part of the work site? (Yes/No)
: Yes

[12] Did the condition occur before or after the contract award?
: Before

[13] Were there affirmatively expressed statements in the contract concerning the relevant subsurface or latent conditions at the site? (Yes/No)
: Yes

There were indications regarding the condition in the contract, therefore, a Type I condition is the most probable avenue of recovery.

[14] Were there latent deficiencies in the contract indications? (Yes/No)
: No

Thinking.......

**** Entitlement will probably not be allowed under DSC claim. ****

I believe that entitlement will probably not be allowed because I concluded that the contractor did fail to-heed contract indications.

The following rule proved to be true causing me to draw the above conclusion:

```
If
'contract indications did not contain latent deficiencies'
                                              is true in answers
      or ('contract information did contain physical data'
                                              is true in answers
         and 'contract information did reveal deficiencies/conditions'
                                              is true in answers)
    then the contractor did fail to-heed contract indications.
```

The following statements contained by the rule are true:

[1] contract indications did not contain latent deficiencies (in answers)

********************* END OF ANALYSIS *********************
(You should receive my bill in 7-10 days.)
(PROMPT PAYMENT IS APPRECIATED)

Do you wish to see the results from this session? (Yes/No)
: Yes

Which of the following results do you wish
to see?

a. The answers given to the questions
b. The unknown information
c. The assumptions which have been made
d. The conclusions which have been reached
e. All of the above
f. None of the above

: e

These are the results of this session:

The answers:
[ANSWERS Database]
CONDITION did occur before CONTRACT AWARD.
CONTRACTOR did not have POTENTIAL-TO-KNOW CONDITION SOONER.
GOVERNMENT did not expressly ASSUME OBLIGATION-RISK.
CONTRACTOR did not expressly ASSUME OBLIGATION-RISK.
CONTRACT did contain STATEMENTS CONCERNING CONDITION.
CONTRACT INDICATIONS did not contain LATENT DEFICIENCIES.
<UNKNOWN> is a date-of-dsc-assertion.
<UNKNOWN> is a date-of-oral-notice.
NOTICE was received-by AUTHORIZED REPRESENTATIVE.
CONDITION is static PHYSICAL PART OF-WORK-SITE.
DSC was asserted.
FINAL-PAYMENT was not made.
FORM-OF-NOTICE was oral.
NOTICE was given before CONTRACTOR DISTURBED CONDITION.
NOTICE was given before REMEDIAL WORK PERFORMED.
CONDITION is directly-related-to-physical-conditions.

The unknowns:
[UNKNOWNS Database]

The assumptions:
[ASSUMPTIONS Database]

The conclusions:
[REPORTS Database]
CONTRACTOR did comply-fully with NOTICE-REQUIREMENT.
REASON_FOR NO ENTITLEMENT does exist.
CONTRACTOR did constructively COMPLY with NOTICE-REQUIREMENT.
CONTRACT did not assign OBLIGATION-RISK.
CONTRACT did contain INDICATIONS CONCERNING CONDITION.
CONTRACTOR did fail TO-HEED CONTRACT INDICATIONS.
NOTICE was given-to RESPONSIBLE RECEIVER.

NOTICE was prompt.

Which of the other results do you want to see?

a. The answers given to the questions
b. The unknown information
c. The assumptions which have been made
d. The conclusions which have been reached
e. All of the above
f. None of the above

: f

Do you wish to store this session? (Yes/No)
: Yes

Please enter your name or initials. (Please place a '-' between initials and/or name.)
: TaK

Session has been saved as requested.

TM
[Rosie   Version 2.3   6-DEC-83 04:50:29]

The name of this session shall be: LAYNE2

Author of previous session: TAK

Date of previous session: 5-DEC-83

You may perform any one of the following analyses:

DSC Assertion   Final Payment   Notice Form
Notice Promptness   Responsible Receiver   Government Prejudiced

Contract Obligation   Excluded Conditions   After-bid Conditions

Express-implied Conditions   Contract Indications
Latent Deficiencies   Standard Conditions
Site Inspection

Superior Knowledge-I   Superior Knowledge-II
Reliance I   Reliance II
Material Difference-I   Material Difference-II
Exculpatory Langauge-I   Exculpatory Language-II

Complete Analysis   No Module

I will begin the analysis as soon as you have selected any ONE
of the options listed. Please enter the selected option
EXACTLY as it appears in the list followed by a '.'.
: Complete Analysis.

SESSION IN PROGRESS

The differing site condition has been asserted.
The date of assertion of the differing site
condition is <UNKNOWN>.

The final payment was not made.

The contractor gave oral notice, therefore,the
contractor constructively complied with the

notice requirement.
The oral notice was given on <UNKNOWN>.

The notice was given before disturbing the condition and before performing the work.
Therefore, notice was prompt.

The contractor should not have been aware of the condition sooner.

Notice of DSC was given to an authorized representative of the CO.

Neither the government nor the contractor have assumed obligation or risk for the condition.

The condition is directly related to the physical conditions at the site.

The condition is a static physical part of the work site.

The condition did occur before the contract was awarded.

The contract did contain statements concerning the relevant subsurface or latent conditions at the site.

There were indications regarding the condition in the contract, therefore, a Type I condition is the most probable avenue of recovery.

There were not latent deficiencies in the contract indications.

Thinking.......

**** Entitlement will probably not be allowed under DSC claim. ****

I believe that entitlement will probably not be allowed because I concluded that the contractor did fail to-heed contract indications.

The following rule proved to be true causing me to draw the above conclusion:

If
'contract indications did not contain latent deficiencies'
is true in answers
or ('contract information did contain physical data'
is true in answers
and 'contract information did reveal deficiencies/conditions'
is true in answers)
then the contractor did fail to-heed contract indications.

The following statements contained by the rule are true:

[1] contract indications did not contain latent deficiencies (in answers)

********************** END OF ANALYSIS **********************
(You should receive my bill in 7-10 days.)
(PROMPT PAYMENT IS APPRECIATED)

Do you wish to see the results from this session? (Yes/No)
: Yes

Which of the following results do you wish to see?

a. The answers given to the questions
b. The unknown information
c. The assumptions which have been made
d. The conclusions which have been reached
e. All of the above
f. None of the above

: e

These are the results of this session:

The answers:
[ANSWERS Database]
CONDITION did occur before CONTRACT AWARD.
CONTRACTOR did not have POTENTIAL-TO-KNOW CONDITION SOONER.

GOVERNMENT did not expressly ASSUME OBLIGATION-RISK.
CONTRACTOR did not expressly ASSUME OBLIGATION-RISK.
CONTRACT did contain STATEMENTS CONCERNING CONDITION.
CONTRACT INDICATIONS did not contain LATENT DEFICIENCIES.
<UNKNOWN> is a date-of-dsc-assertion.
<UNKNOWN> is a date-of-oral-notice.
NOTICE was received-by AUTHORIZED REPRESENTATIVE.
CONDITION is static PHYSICAL PART OF-WORK-SITE.
DSC was asserted.
FINAL-PAYMENT was not made.
FORM-OF-NOTICE was oral.
NOTICE was given before CONTRACTOR DISTURBED CONDITION.
NOTICE was given before REMEDIAL WORK PERFORMED.
CONDITION is directly-related-to-physical-conditions.

The unknowns:
[UNKNOWNS Database]

The assumptions:
[ASSUMPTIONS Database]

The conclusions:
[REPORTS Database]
CONTRACTOR did comply-fully with NOTICE-REQUIREMENT.
REASON_FOR NO ENTITLEMENT does exist.
CONTRACTOR did constructively COMPLY with NOTICE-REQUIREMENTS.
GOVERNMENT did receive NOTICE-OF-CONDITION.
CONTRACT did not assign OBLIGATION-RISK.
CONTRACT did contain INDICATIONS CONCERNING CONDITION.
CONTRACTOR did fail TO-HEED CONTRACT INDICATIONS.
NOTICE was prompt.
NOTICE was given to RESPONSIBLE RECEIVER.

Which of the other results do you want to see?

a. The answers given to the questions
b. The unknown information
c. The assumptions which have been made
d. The conclusions which have been reached
e. All of the above
f. None of the above

: f

Do you wish to store this session? (Yes/No)
: No

Session NOT saved.

75-1 BCA

[¶ 11,048] C.H. LEAVELL & CO.

ASBCA No. 18625, January 15, 1975.

Contract No. DACA63-C0060.

Changed Conditions--Subsurface Conditions--Reliability of Government Supplied Information--Duty to Inquire.

A construction contractor was not entitled to an equitable adjustment for the additional expenses he allegedly incurred due to an unexpected amount of water seepage into his foundation borings, even though the government furnished data concerning underground water was conflictive, because the contractor failed to seek additional information concerning the conflicts. Although the government's test borings indicated that the subsurface soil was practically impervious to water, some of the drawing symbols which accompanied the government data were unclear and not the same as those symbols contained in the military standards. The contractor contended that since some of the symbols indicated that the soil was impervious to water, those that were unknown could also have been interpreted as being impervious soil. The contractor's decision to assume that the subsurface water would not enter the holes for a period of two hours after drilling was not founded on a reasonable interpretation of the information provided. Since he desired to reach such a conclusion on the basis of imprecise data provided by the government, he had a duty to

seek clarification from the government and, absent such a clarification, he proceeded at his own risk. The motion for reconsideration of the prior decision (74-2 BCA ¶ 10,885) was therefore denied.

74-2 BCA

[¶ 10,885] LEAVELL & CO., C.H.

ASBCA No. 18625. October 15, 1974.

Contract No. DACA63-68-C-0060.

Changed Conditions--Subsurface Conditions--Reliability of Government Information.

A construction contractor was not entitled to an equitable adjustment for additional expenses incurred due to an unexpected amount of water in the soil because the government's test boring information furnished to the contractor indicated that the contractor would encounter a significant amount of water while constructing the building. It was the contractor's position that the subsurface conditions at the site were materially different from those conditions represented in the bidding and contracting documents. Although the government's borings indicated that there would be some water encountered in every hole sunk, the contractor's two borings indicated very little water. The contractor, however, did not provide any persuasive evidence as to how he could have accepted the government's findings without seeking further information. The data presented by the government was inconsistent and confusing and, therefore, a reasonably prudent contractor would have sought clarification. Absent such clarification, he could not hold the government responsible.

TM
[Rosie Version 2.3 5-DEC-83 02:50:23]

The name of this session shall be: LEAVELL

You may perform any one of the following analyses:

DSC Assertion Final Payment Notice Form
Notice Promptness Responsible Receiver Government Prejudiced

Contract Obligation Excluded Conditions After-bid Conditions

Express-implied Conditions Contract Indications
Latent Deficiencies Standard Conditions
Site Inspection

Superior Knowledge-I Superior Knowledge-II
Reliance I Reliance II
Material Difference-I Material Difference-II
Exculpatory Langauge-I Exculpatory Language-II

Complete Analysis No Module

I will begin the analysis as soon as you have selected any ONE of the options listed. Please enter the selected option EXACTLY as it appears in the list followed by a '.'.
: Complete Analysis.

SESSION IN PROGRESS

[1] Has a differing site condition been asserted by one of your contractors? (Yes/No)
: Yes

[2] On what date did the DSC-ASSERTION occur? (month day, year)
: July 10, 1969

[3] Has final payment been made? (Yes/No)
: No

[4] What form of notice was given to the government by the contractor? (Select one of the following.)
a. Written notice was given.
b. Oral notice was given.
c. Other events surrounded notice.

: a

[5] On what date did the WRITTEN-NOTICE occur? (month day, year)
: July 10, 1969

[6] When was notice given in relation to the progression xD
of work?
a. Notice given before contractor disturbed the condition.
b. Notice given before contractor performed remedial work.
c. Both a and b.
d. None of the above.
: d

[7] Is there evidence that the contractor should have been
aware of the differing site condition sooner? (Yes/No)
: Yes

[8] To whom was notice of the differing site condition given?
a. The contracting officer
b. An authorized representative of the contracting officer
c. Any other employee of the government
d. None of the above
: a

[9] Is the government able to show that evidence against
the claim was obscurred due to the passage of time? (Yes/No)
: Yes

[10] Is the government able to show that proper notice
would have resulted in cheaper resolution of the condition
by CO? (Yes/No)
: No

[11] Can the government show that it suffered any other prejudice
through inadequacy of or lack of notice? (Yes/No)
: -

The government did suffer prejudice.

[12] Was the government defense made impossible by inadequacy
of or lack of notice? (Yes/No)
: No

[13] Was the contractor able to provide sufficient additional proof of claim sufficient to prove entitlement to DSC claim in spite of slight prejudice to the government? (Yes/No)
: N-

[14] Who expressly assumed obligation for or risk of the condition in the contract?
a. The government
b. The contractor
c. Both the government and the contractor
d. Neither the government nor the contractor
: d

[15] Is the condition in question directly related to the physical conditions at the work site? (Yes/No)
: Yes

[16] Is the condition a static physical part of the work site? (Yes/No)
: Yes

[17] Did the condition occur before or after the contract award?
: Before

[18] Were there affirmatively expressed statements in the contract concerning the relevant subsurface or latent conditions at the site? (Yes/No)
: Yes

There were indications regarding the condition in the contract, therefore, a Type I condition is the most probable avenue of recovery.

[19] Were there latent deficiencies in the contract indications? (Yes/No)
: Yes

[20] Was the physical data made a part of the contract? (Yes/No)
: Yes

[21] Would this physical data have revealed or resolved the apparent deficiencies in the contract indications for a reasonable contractor? (Yes/No)
: -

[22] Did the GOVERNMENT know about the CONTRACT-DEFICIENCY? (Yes/No)
: Yes

[23] Did the government reveal CONTRACT-DEFICIENCY to the contractor? (Yes/No)
: No

Oversight, misrepresentation or concealment on the part of the government. A possible breach has occurred. However, I will continue with the analysis.

[24] Was the contractor in possession of unilateral superior or imputed knowledge that the actual conditions patently differed from the indications in the contract? (Yes/No)
: No

[25] Would simple inquiries by the contractor have revealed the condition to be contrary to the CONTRACT-INDICATIONS? (Yes/No)
: Yes

It has become obvious that the contractor did not make simple inquiries.

Thinking.......

**** Entitlement will probably not be allowed under DSC claim. ****

I believe that entitlement will probably not be allowed because I
concluded that the contractor did not make simple inquiries.

The following rule proved to be true causing me
to draw the above conclusion:

If
'simple inquiry did give potential to-know contrary conditions'
is true in answers
then the contractor did not make simple inquiries.

The following statements contained by the rule
are true:

[1] simple inquiry did give potential to-know contrary conditions (in answers
)

********************* END OF ANALYSIS *********************
(You should receive my bill in 7-10 days.)
(PROMPT PAYMENT IS APPRECIATED)

Do you wish to see the results from this session? (Yes/No)
: Yes

Which of the following results do you wish
to see?

a. The answers given to the questions
b. The unknown information
c. The assumptions which have been made
d. The conclusions which have been reached
e. All of the above
f. None of the above

: e

These are the results of this session:

The answers:
[ANSWERS Database]
CONDITION did occur before CONTRACT AWARD.
CONTRACTOR did have POTENTIAL-TO-KNOW CONDITION SOONER.
PASSAGE-OF-TIME did obscure EVIDENCE.
IMPROPER-NOTICE did not cause ADDITIONAL COST.
GOVERNMENT did not expressly ASSUME OBLIGATION-RISK.
CONTRACTOR did not expressly ASSUME OBLIGATION-RISK.
CONTRACT did contain STATEMENTS CONCERNING CONDITION.
CONTRACT INDICATIONS did contain LATENT DEFICIENCIES.

CONTRACT INFORMATION did contain PHYSICAL DATA.
GOVERNMENT did not reveal CONTRACT-DEFICIENCY.
SIMPLE INQUIRY did give POTENTIAL TO-KNOW CONTRARY CONDITIONS.
<JULY 10, 1969> is a date-of-dsc-assertion.
<JULY 10, 1969> is a date-of-written-notice.
NOTICE was received-by CONTRACTING OFFICER.
DEFENSE-AGAINST CLAIM was not made IMPOSSIBLE.
GOVERNMENT was aware-of CONTRACT-DEFICIENCY.
CONTRACTOR was not aware-of CONTRACT-DEFICIENCY.
CONDITION is static PHYSICAL PART OF-WORK-SITE.
DSC was asserted.
FINAL-PAYMENT was not made.
FORM-OF-NOTICE was written.
NOTICE was not given before REMEDIAL WORK PERFORMED.
CONDITION is directly-related-to-physical-conditions.

The unknowns:
[UNKNOWNS Database]
IF-ANYTHING-ELSE did cause PREJUDICE.
IF-CONTRACT INFORMATION did reveal DEFICIENCIES/CONDITIONS.
IF-ADDITIONAL-PROOF does exist TO-PROVE-ENTITLEMENT.

The assumptions:
[ASSUMPTIONS Database]
NOTHING-ELSE did cause PREJUDICE.
CONTRACT INFORMATION did not reveal DEFICIENCIES/CONDITIONS.
ADDITIONAL-PROOF does exist TO-PROVE-ENTITLEMENT.

The conclusions:
[REPORTS Database]
CONTRACTOR did comply with WRITTEN-NOTICE-REQUIREMENT.
REASON_FOR NO ENTITLEMENT does exist.
GOVERNMENT did receive NOTICE-OF-CONDITION.
PASSAGE-OF-TIME did cause PREJUDICE.
GOVERNMENT did suffer PREJUDICE.
CONTRACT did not assign OBLIGATION-RISK.
CONTRACT did contain INDICATIONS CONCERNING CONDITION.
GOVERNMENT did have SUPERIOR KNOWLEDGE.
CONTRACTOR did not make SIMPLE INQUIRIES.
PROMPTNESS_OF_NOTICE is an in_doubt.
NOTICE was given-to RESPONSIBLE RECEIVER.

Which of the other results do you want to see?

a. The answers given to the questions
b. The unknown information
c. The assumptions which have been made
d. The conclusions which have been reached
e. All of the above

f. None of the above
: f

Do you wish to store this session? (Yes/No)
: Yes

Please enter your name or initials. (Please place a '-' between initials and/or name.)
: TaK

Session has been saved as requested.

77-1 BCA

[¶ 12,335] BERNARD McMENAMY CONTRACTOR, INC.

ENG BCA No. 3413. December 22, 1976.

Contract No. DACW03-68-C-0048.

Changed Conditions--Rock, Water and Other Subsurface Conditions--Rock and Stone--Quality Variations.

A dredging contractor was entitled to an equitable adjustment for removing large quantities of rock buried in the channel bed which interfered with dredge operations because site inspection, core borings, and contract specifications did not apprise him of the size and quantity of rock actually encountered. A meaningful site inspection could not have been performed because the river was near flood stage during the time available for bidding. Contract drawings and specifications indicated that the material to be dredged was mostly sand and gravel. Core borings showed silt, sand, and gravel, but noted the existence of boulders below the channel bottom in one area. The contractor encountered, however, substantial but varying amounts of rock, cobbles, and boulders throughout the dredging area. The government contended that the contract's Character of Materials clause warned that all size material would be encountered. This clause meant only normal variations between the core borings and the subsurface conditions should be expected; it did not relieve the

government of liability for a variation of the magnitude the contractor encountered.

TM
[Rosie Version 2.3 5-DEC-83 00:50:36]

The name of this session shall be: McMENAM

You may perform any one of the following analyses:

DSC Assertion Final Payment Notice Form
Notice Promptness Responsible Receiver Government Prejudiced

Contract Obligation Excluded Conditions After-bid Conditions

Express-implied Conditions Contract Indications
Latent Deficiencies Standard Conditions
Site Inspection

Superior Knowledge-I Superior Knowledge-II
Reliance I Reliance II
Material Difference-I Material Difference-II
Exculpatory Langauge-I Exculpatory Language-II

Complete Analysis No Module

I will begin the analysis as soon as you have selected any ONE
of the options listed. Please enter the selected option
EXACTLY as it appears in the list followed by a '.'.
: Complete Analysis.

SESSION IN PROGRESS

[1] Has a differing site condition been asserted by one of
your contractors? (Yes/No)
: Yes

[2] On what date did the DSC-ASSERTION occur? (month day, year)
: -

[3] Has final payment been made? (Yes/No)
: No

[4] What form of notice was given to the government by
the contractor? (Select one of the following.)
a. Written notice was given.
b. Oral notice was given.
c. Other events surrounded notice.

: b

[5] On what date did the ORAL-NOTICE occur? (month day, year)
: -

[6] When was notice given in relation to the progression of work?
a. Notice given before contractor disturbed the condition.
b. Notice givK efore contractor performed remedial work.
c. Both a and b.
d. None of the above.
: d

[7] Is there evidence that the contractor should have been aware of the differing site condition sooner? (Yes/No)
: -

[8] To whom was notice of the differing site condition given?
a. The contracting officer
b. An authorized representative of the contracting officer
c. Any other employee of the government
d. None of the above
: c

[9] Was this other government employee cognisant of the implications of the condition? (Yes/No)
: Yes

[10] Did this other government employee communicate his awareness of the DSC to the CO? (Yes/No)
: Yes

[11] Is the government able to show that evidence against the claim was obscurred due to the passage of time? (Yes/No)
: No

[12] Is the government able to show that proper notice would have resulted in cheaper resolution of the condition by CO? (Yes/No)
: No

[13] Can the government show that it suffered any other prejudice through inadequacy of or lack of notice? (Yes/No)

: No

Government can't prove that it suffered prejudice.

[14] Who expressly assumed obligation for or risk of the condition in the contract?
a. The government
b. The contractor
c. Both the government and the contractor
d. Neither the government nor the contractor
: d

[15] Is the condition in question directly related to the physical conditions at the work site? (Yes/No)
: Yes

[16] Is the condition a static physical part of the work site? (Yes/No)
: Yes

[17] Did the condition occur before or after the contract award?
: Before

[18] Were there affirmatively expressed statements in the contract concerning the relevant subsurface or latent conditions at the site? (Yes/No)
: Yes

There were indications regarding the condition in the contract, therefore, a Type I condition is the most probable avenue of recovery.

[19] Were there latent deficiencies in the contract indications? (Yes/No)
: Yes

[20] Was the physical data made a part of the

contract? (Yes/No)
: Yes

[21] Would this physical data have revealed or resolved the apparent deficiencies in the contract indications for a reasonable contractor? (Yes/No)
: No

[22] Did the GOVERNMENT know about the CONTRACT-DEFICIENCY? (Yes/No)
: -

[23] Was the contractor in possession of unilateral superior or imputed knowledge that the actual conditions patently differed from the indications in the contract? (Yes/No)
: No

[24] Would simple inquiries by the contractor have revealed the condition to be contrary to the CONTRACT-INDICATIONS? (Yes/No)
: No

[25] Did any of the following acts of the government hinder the site inspection?

a. Access to site was denied
b. Inadequate time was allowed
c. No act of the government hindered inspection
: -

[26] Did the contractor conduct a site inspection? (Yes/No)
: -

[27] Would the actual condition have been discernable by a layman contractor performing a reasonable inspection? (Yes/No)
: -

[28] Did the contractor reasonably rely on DEFICIENT-INDICATIONS-IN-CONTRACT when preparing his bid causing him to suffer prejudice

through this reliance? (Yes/No)
: Yes

[29] Was the difference between actual and INDICATED conditions a material difference? (Yes/No)
: Yes

Relief is probably available through the Type I DSC clause because of deficient indications in the contract.

[30] Is there an exculpatory clause denying any government liability and responsibility for actual conditions different from those indicated in the contract documents? (Yes/No)
: Yes

[31] Is the exculpatory clause specific to the DSC clause and to the condition encountered? (Yes/No)
: No

[32] Are the language and intent of the exculpatory clause clear and unambiguous? (Yes/No)
: No

I believe that the exculpatory clause is probably not valid for this case.

Therefore, entitlement will probably be allowed. Consult your local CO for further details.

Thinking.......

********************** END OF ANALYSIS **********************
(You should receive my bill in 7-10 days.)
(PROMPT PAYMENT IS APPRECIATED)

Do you wish to see the results from this session? (Yes/No)
: Yes

Which of the following results do you wish
to see?

a. The answers given to the questions
b. The unknown information
c. The assumptions which have been made
d. The conclusions which have been reached
e. All of the above
f. None of the above

: e

These are the results of this session:

The answers:
[ANSWERS Database]
CONDITION did occur before CONTRACT AWARD.
GOVERNMENT EMPLOYEE did understand IMPLICATIONS.
GOVERNMENT EMPLOYEE did communicate DSC AWARENESS TO-CO.
PASSAGE-OF-TIME did not obscure EVIDENCE.
IMPROPER-NOTICE did not cause ADDITIONAL COST.
NOTHING-ELSE did cause PREJUDICE.
GOVERNMENT did not expressly ASSUME OBLIGATION-RISK.
CONTRACTOR did not expressly ASSUME OBLIGATION-RISK.
CONTRACT did contain STATEMENTS CONCERNING CONDITION.
CONTRACT INDICATIONS did contain LATENT DEFICIENCIES.
CONTRACT INFORMATION did contain PHYSICAL DATA.
CONTRACT INFORMATION did not reveal DEFICIENCIES/CONDITIONS.
SIMPLE INQUIRY did not give POTENTIAL TO-KNOW CONTRARY CONDITIONS.
CONTRACTOR did suffer PREJUDICE through RELIANCE.
CONTRACT does contain EXCULPATORY-CLAUSE.
<UNKNOWN> is a date-of-dsc-assertion.
<UNKNOWN> is a date-of-oral-notice.
NOTICE was received-by OTHER GOVERNMENT EMPLOYEE.
CONTRACTOR was not aware-of CONTRACT-DEFICIENCY.
CONDITION is static PHYSICAL PART OF-WORK-SITE.
EXCULPATORY-CLAUSE is not specific-to DSC CLAUSE.
DSC was asserted.
FINAL-PAYMENT was not made.
FORM-OF-NOTICE was oral.
NOTICE was not given before REMEDIAL WORK PERFORMED.
DIFFERENCE was material.
CONDITION is directly-related-to-physical-conditions.
EXCULPATORY-CLAUSE is not clear-and-unambiguous.

The unknowns:
[UNKNOWNS Database]
IF-CONTRACTOR did have POTENTIAL-TO-KNOW CONDITION SOONER.
IF-REASONABLE INSPECTION did have POTENTIAL-TO-REVEAL CONDITION.
IF-GOVERNMENT did hinder INSPECTION.
IF-CONTRACTOR did conduct SITE-INSPECTION.
IF-GOVERNMENT was aware-of CONTRACT-DEFICIENCY.

The assumptions:
[ASSUMPTIONS Database]
CONTRACTOR did not have POTENTIAL-TO-KNOW CONDITION SOONER.
REASONABLE INSPECTION did not have POTENTIAL-TO-REVEAL CONDITION.
GOVERNMENT did not hinder INSPECTION.
CONTRACTOR did conduct SITE-INSPECTION.
GOVERNMENT was not aware-of CONTRACT-DEFICIENCY.

The conclusions:
[REPORTS Database]
CONTRACTOR did constructively COMPLY with NOTICE-REQUIREMENT.
GOVERNMENT did receive NOTICE-OF-CONDITION.
GOVERNMENT did not suffer PREJUDICE.
CONTRACT did not assign OBLIGATION-RISK.
CONTRACT did contain INDICATIONS CONCERNING CONDITION.
PROMPTNESS_OF_NOTICE is an in_doubt.
NOTICE was given-to RESPONSIBLE RECEIVER.
CONTRACTOR is not required-to INSPECT OTHER DATA.
EXCULPATORY-CLAUSE is not probably VALID.
NOTICE was prompt.
REASONABLE INSPECTION is not required-for-entitlement.
VALIDITY-OF EXCULPATORY-CLAUSE is in_doubt.
ENTITLEMENT is still-probable.

Which of the other results do you want to see?

a. The answers given to the questions
b. The unknown information
c. The assumptions which have been made
d. The conclusions which have been reached
e. All of the above
f. None of the above

: f

Do you wish to store this session? (Yes/No)
: Yes

Please enter your name or initials. (Please place a '-'
between initials and/or name.)
: TaK

Session has been saved as requested.

77-1 BCA

[¶ 12,225] NORAIR ENGINEERING CORPORATION

ENG BCA No. 3568. April 30, 1976.

Contract No. 1B0021.

Changes--Rock, Water and Other Subsurface Conditions--Flow from Previously Open Excavations.

A contractor was entitled to an equitable adjustment when groundwater and sewage flowed into his excavation because the flow was unusual and unexpected. The excavation had been open for three or four months before the flow commenced. Although the government claimed that the flow was groundwater, it made no difference because nothing in the contract put the contractor on notice that such a flow might be encountered. Further, the elevation of the invert of the sewer line was not clearly established and there was no explanation why groundwater would flow through or along the sewer if its invert was two feet below the top of the groundwater table.

TM
[Rosie Version 2.3 2-DEC-83 11:50:42]

The name of this session shall be: NORAIR

You may perform any one of the following analyses:

DSC Assertion   Final Payment   Notice Form
Notice Promptness   Responsible Receiver   Government Prejudiced

Contract Obligation   Excluded Conditions   After-bid Conditions

Express-implied Conditions   Contract Indications
Latent Deficiencies   Standard Conditions
Site Inspection

Superior Knowledge-I   Superior Knowledge-II
Reliance I   Reliance II
Material Difference-I   Material Difference-II
Exculpatory Langauge-I   Exculpatory Language-II

Complete Analysis   No Module

I will begin the analysis as soon as you have selected any ONE
of the options listed. Please enter the selected option
EXACTLY as it appears in the list followed by a '.'.
: Complete Analysis.

SESSION IN PROGRESS

[1] Has a differing site condition been asserted by one of
your contractors? (Yes/No)
: Yes

[2] On what date did the DSC-ASSERTION occur? (month day, year)
: April 19, 1973

[3] Has final payment been made? (Yes/No)
: No

[4] What form of notice was given to the government by
the contractor? (Select one of the following.)

a. Written notice was given.
b. Oral notice was given.
c. Other events surrounded notice.

: a

[5] On what date did the WRITTEN-NOTICE occur? (month day, year)
: April 19, 1973

[6] When was notice given in relation to the progression of work?
a. Notice given before contractor disturbed the condition.
b. Notice given before contractor performed remedial work.
c. Both a and b.
: c

[7] Is there evidence that the contractor should have been aware of the differing site condition sooner? (Yes/No)
: No

[8] To whom was notice of the differing site condition given?
a. The contracting officer
b. An authorized representative of the contracting officer
c. Any other employee of the government
d. None of the above
: a

[9] Who expressly assumed obligation for or risk of the condition in the contract?
a. The government
b. The contractor
c. Both the government and the contractor
d. Neither the government nor the contractor
: -

[10] Is the condition in question directly related to the physical conditions at the work site? (Yes/No)
: Yes

[11] Is the condition a static physical part of the work site? (Yes/No)
: Yes

[12] Did the condition occur before or after the contract award?
: Before

[13] Were there affirmatively expressed statements in the contract concerning the relevant subsurface or latent conditions at the site? (Yes/No)
: No

[14] Would the affirmatively expressed contract statements on the general conditions at the site lead a reasonable contractor to believe that the condition could be expected? (Yes/No)
: No

[15] Were indications of subsurface or latent conditions inferred from the contract as a whole? (Yes/No)
: No

There were no indications regarding the condition in the contract, therefore, a Type II condition is the most probable avenue of recovery.

[16] Was the physical data made a part of the contract? (Yes/No)
: Yes

[17] Would this physical data have indicated the nature of existing conditions to a reasonable contractor? (Yes/No)
: No

[18] By what means can the contractor establish expected conditions?
a. Customs of the trade
b. General knowledge in the industry
c. Manufacturers instructions or recommendations
d. None of the above
: b

[19] Does the actual condition differ from the standard? (Yes/No)
: Yes

[20] Did the GOVERNMENT know about the ACTUAL-CONDITIONS? (Yes/No)
: -

[21] Was the contractor in possession of unilateral superior knowledge of the actual conditions? (Yes/No)
: No

[22] Would simple inquiries by the contractor have revealed the condition to be contrary to the STANDARD-OF-EXPECTATION? (Yes/No)
: No

[23] Did any of the following acts of the government hinder the site inspection?

a. Access to site was denied
b. Inadequate time was allowed
c. No act of the government hindered inspection
: c

[24] Did the contractor conduct a site inspection? (Yes/No)
: -

[25] Would the actual condition have been discernable by a layman contractor performing a reasonable inspection? (Yes/No)
: No

[26] Did the contractor reasonably rely on EXPECTATIONS-THAT-CONDITION-WOULD-MEET-STANDARD when preparing his bid causing him to suffer prejudice through this reliance? (Yes/No)
: Yes

[27] Was the difference between actual and EXPECTED conditions a material difference? (Yes/No)
: Yes

Relief is probably available through the Type II DSC clause because of unanticipated or unusual conditions or materials.

[28] Is there a clause in the contract placing the risk of conditions not revealed by the contractor's site

inspection? (Yes/No)
: Yes

[29] Is the exculpatory clause specific to the DSC clause and to the condition encountered? (Yes/No)
: No

[30] Are the language and intent of the exculpatory clause clear and unambiguous? (Yes/No)
: -
CLAUSECLAUSESince the exculpatory language is not specific to the DSC clause there is only a 50/50 chance that entitlement will be justified.

Consult your local CO for further details.

Thinking.......

********************** END OF ANALYSIS **********************
(You should receive my bill in 7-10 days.)
(PROMPT PAYMENT IS APPRECIATED)

Do you wish to see the results from this session? (Yes/No)
: Yes

Which of the following results do you wish to see?

a. The answers given to the questions
b. The unknown information
c. The assumptions which have been made
d. The conclusions which have been reached
e. All of the above
f. None of the above

: e

These are the results of this session:

The answers:
[ANSWERS Database]
CONDITION did occur before CONTRACT AWARD.
CONDITION does differ from STANDARD-OF-EXPECTATION.
CONTRACTOR did not have POTENTIAL-TO-KNOW CONDITION SOONER.
REASONABLE INSPECTION did not have POTENTIAL-TO-REVEAL CONDITION.
CONTRACT did not contain STATEMENTS CONCERNING CONDITION.
CONTRACT did not contain GENERAL CONDITIONS IMPLYING CONDITION.

CONTRACT INFORMATION did contain PHYSICAL DATA.
CONTRACT did not lead ONE-TO INFER CONDITIONS.
CONTRACT INFORMATION did not reveal DEFICIENCIES/CONDITIONS.
SIMPLE INQUIRY did not give POTENTIAL TO-KNOW CONTRARY CONDITIONS.
GOVERNMENT did not hinder INSPECTION.
CONTRACTOR did suffer PREJUDICE through RELIANCE.
CONTRACT does contain EXCULPATORY-CLAUSE.
<APRIL 19, 1973> is a date-of-dsc-assertion.
<APRIL 19, 1973> is a date-of-written-notice.
NOTICE was received-by CONTRACTING OFFICER.
CONTRACTOR was not aware-of ACTUAL-CONDITIONS.
CONDITION is static PHYSICAL PART OF-WORK-SITE.
EXCULPATORY-CLAUSE is not specific-to DSC CLAUSE.
DSC was asserted.
FINAL-PAYMENT was not made.
FORM-OF-NOTICE was written.
NOTICE was given before CONTRACTOR DISTURBED CONDITION.
NOTICE was given before REMEDIAL WORK PERFORMED.
DIFFERENCE was material.
CONDITION is directly-related-to-physical-conditions.
EXPECTED CONDITIONS is establishable from GENERAL-KNOWLEDGE-IN-INDUSTRY

The unknowns:
[UNKNOWNS Database]
WHICH-PARTY did assume RISK.
IF-CONTRACTOR did conduct SITE-INSPECTION.
IF-GOVERNMENT was aware-of ACTUAL-CONDITIONS.
IF-EXCULPATORY-CLAUSE is clear-and-unambiguous.

The assumptions:
[ASSUMPTIONS Database]
GOVERNMENT did not expressly ASSUME OBLIGATION-RISK.
CONTRACTOR did not expressly ASSUME OBLIGATION-RISK.
CONTRACTOR did conduct SITE-INSPECTION.
GOVERNMENT was not aware-of ACTUAL-CONDITIONS.
EXCULPATORY-CLAUSE is clear-and-unambiguous.

The conclusions:
[REPORTS Database]
CONTRACTOR did comply with WRITTEN-NOTICE-REQUIREMENT.
CONTRACTOR did comply-fully with NOTICE-REQUIREMENT.
GOVERNMENT did receive NOTICE-OF-CONDITION.
CONTRACT did not contain INDICATIONS CONCERNING CONDITION.
NOTICE was given-to RESPONSIBLE RECEIVER.
CONTRACTOR is not required-to INSPECT OTHER DATA.
CONTRACTOR is able-to ESTABLISH STANDARD-OF-EXPECTATION.
NOTICE was prompt.
REASONABLE INSPECTION is not required-for-entitlement.
VALIDITY-OF EXCULPATORY-CLAUSE is in_doubt.

ENTITLEMENT is still-probable.

Which of the other results do you want to see?

a. The answers given to the questions
b. The unknown information
c. The assumptions which have been made
d. The conclusions which have been reached
e. All of the above
f. None of the above

: f

Do you wish to store this session? (Yes/No)
: Yes

Please enter your name or initials. (Please place a '-' between initials and/or name.)
: TAK

Session has been saved as requested.

75-1 BCA

[¶ 11,274] JACK PICOULT

VACAB No. 1095. May 28, 1975.

Contract No. V1006C-763.

Changed Conditions--Differing Site Conditions Clause--Differing Subsurface or Latent Physical Conditions.

A construction contractor was entitled to an equitable adjustment for miscalculations in his bid because the government's positive representations of site conditions in the contract documents proved erroneous. The government relied upon the general requirements of the contract, which placed the responsibility for performing all work that could reasonably have been anticipated by study of the bidding documents and by inspection of the work site on the contractor. The government claimed that the requirement in the IFB specifications which required the bidders to inspect the site before submitting bids limited government liability for claims under the Differing Site Conditions clause. However, the contractor was entitled to rely on the government's positive representations of site conditions which later proved faulty, and did not have to make independent investigations of those particular points. Furthermore, the site investigation requirement did not nullify the Differing Site Conditions clause and did not obligate the contractor to discover latent conditions

or conditions which were not apparent from a reasonable investigation.

TM
[Rosie Version 2.3 5-DEC-83 02:04:29]

The name of this session shall be: PICOULT

You may perform any one of the following analyses:

DSC Assertion Final Payment Notice Form
Notice Promptness Responsible Receiver Government Prejudiced

Contract Obligation Excluded Conditions After-bid Conditions

Express-implied Conditions Contract Indications
Latent Deficiencies Standard Conditions
Site Inspection

Superior Knowledge-I Superior Knowledge-II
Reliance I Reliance II
Material Difference-I Material Difference-II
Exculpatory Langauge-I Exculpatory Language-II

Complete Analysis No Module

I will begin the analysis as soon as you have selected any ONE
of the options listed. Please enter the selected option
EXACTLY as it appears in the list followed by a '.'.
: Complete Analysis.

SESSION IN PROGRESS

[1] Has a differing site condition been asserted by one of
your contractors? (Yes/No)
: Yes

[2] On what date did the DSC-ASSERTION occur? (month day, year)
: -

[3] Has final payment been made? (Yes/No)
: No

[4] What form of notice was given to the government by
the contractor? (Select one of the following.)

a. Written notice was given.
b. Oral notice was given.
c. Other events surrounded notice.

: a

[5] On what date did the WRITTEN-NOTICE occur? (month day, year)
: October 21, 1971

[6] When was notice given in relation to the progression of work?
a. Notice given before contractor disturbed the condition.
b. Notice given before contractor performed remedial work.
c. Both a and b.
d. None of the above.
: c

[7] Is there evidence that the contractor should have been aware of the differing site condition sooner? (Yes/No)
: -

[8] To whom was notice of the differing site condition given?
a. The contracting officer
b. An authorized representative of the contracting officer
c. Any other employee of the government
d. None of the above
: a

[9] Who expressly assumed obligation for or risk of the condition in the contract?
a. The government
b. The contractor
c. Both the government and the contractor
d. Neither the government nor the contractor
: d

[10] Is the condition in question directly related to the physical conditions at the work site? (Yes/No)
: Yes

[11] Is the condition a static physical part of the work site? (Yes/No)
: Yes

[12] Did the condition occur before or after the contract award?
: Before

[13] Were there affirmatively expressed statements in the contract concerning the relevant subsurface or latent conditions at the site? (Yes/No)
: Yes

There were indications regarding the condition in the contract, therefore, a Type I condition is the most probable avenue of recovery.

[14] Were there latent deficiencies in the contract indications? (Yes/No)
: Yes

[15] Was the physical data made a part of the contract? (Yes/No)
· No

[16] Did the GOVERNMENT know about the CONTRACT-DEFICIENCY? (Yes/No)
: No

[17] Was the contractor in possession of unilateral superior or imputed knowledge that the actual conditions patently differed from the indications in the contract? (Yes/No)
: No

[18] Would simple inquiries by the contractor have revealed the condition to be contrary to the CONTRACT-INDICATIONS? (Yes/No)
: No

[19] Did any of the following acts of the government hinder the site inspection?

a. Access to site was denied
b. Inadequate time was allowed
c. No act of the government hindered inspection
: c

[20] Did the contractor conduct a site inspection? (Yes/No)
: Yes

[21] Was the inspection conducted by the contractor a reasonable inspection, i.e., one comparable to that expected from a reasonable, prudent contractor experienced in that particular field of work? (Yes/No)
: Yes

[22] Would the actual condition have been discernable by a layman contractor performing a reasonable inspection? (Yes/No)
: No

[23] Did the contractor reasonably rely on DEFICIENT-INDICATIONS-IN-CONTRACT when preparing his bid causing him to suffer prejudice through this reliance? (Yes/No)
: Yes

[24] Was the difference between actual and INDICATED conditions a material difference? (Yes/No)
: Yes

Relief is probably available through the Type I DSC clause because of deficient indications in the contract.

[25] Is there an exculpatory clause denying any government liability and responsibility for actual conditions different from those indicated in the contract documents? (Yes/No)
: No

Since the contract does not contain an exculpatory clause entitlement will probably be allowed.

Consult your local CO for further details.

Thinking.......

******************** END OF ANALYSIS ********************
(You should receive my bill in 7-10 days.)
(PROMPT PAYMENT IS APPRECIATED)

Do you wish to see the results from this session? (Yes/No)
: Yes

Which of the following results do you wish
to see?

a. The answers given to the questions
b. The unknown information
c. The assumptions which have been made
d. The conclusions which have been reached
e. All of the above
f. None of the above

: e

These are the results of this session:

The answers:
[ANSWERS Database]
CONDITION did occur before CONTRACT AWARD.
GOVERNMENT did not expressly ASSUME OBLIGATION-RISK.
CONTRACTOR did not expressly ASSUME OBLIGATION-RISK.
CONTRACT did contain STATEMENTS CONCERNING CONDITION.
CONTRACT INDICATIONS did contain LATENT DEFICIENCIES.
CONTRACT INFORMATION did not contain PHYSICAL DATA.
SIMPLE INQUIRY did not give POTENTIAL TO-KNOW CONTRARY CONDITIONS.
GOVERNMENT did not hinder INSPECTION.
CONTRACTOR did conduct SITE-INSPECTION.
REASONABLE INSPECTION did not have POTENTIAL-TO-REVEAL CONDITION.
CONTRACTOR did suffer PREJUDICE through RELIANCE.
CONTRACT does not contain EXCULPATORY-CLAUSE.
<UNKNOWN> is a date-of-dsc-assertion.
<OCTOBER 21, 1971> is a date-of-written-notice.
NOTICE was received-by CONTRACTING OFFICER.
GOVERNMENT was not aware-of CONTRACT-DEFICIENCY.
CONTRACTOR was not aware-of CONTRACT-DEFICIENCY.
SITE-INSPECTION was reasonable INSPECTION.
CONDITION is static PHYSICAL PART OF-WORK-SITE.
DSC was asserted.
FINAL-PAYMENT was not made.
FORM-OF-NOTICE was written.
NOTICE was given before CONTRACTOR DISTURBED CONDITION.
NOTICE was given before REMEDIAL WORK PERFORMED.
DIFFERENCE was material.
CONDITION is directly-related-to-physical-conditions.

The unknowns:

[UNKNOWNS Database]
IF-CONTRACTOR did have POTENTIAL-TO-KNOW CONDITION SOONER.

The assumptions:
[ASSUMPTIONS Database]
CONTRACTOR did not have POTENTIAL-TO-KNOW CONDITION SOONER.

The conclusions:
[REPORTS Database]
CONTRACTOR did comply with WRITTEN-NOTICE-REQUIREMENT.
CONTRACTOR did comply-fully with NOTICE-REQUIREMENT.
GOVERNMENT did receive NOTICE-OF-CONDITION.
CONTRACT did not assign OBLIGATION-RISK.
CONTRACT did contain INDICATIONS CONCERNING CONDITION.
REASONABLE INSPECTION did not reveal CONDITION.
NOTICE was given-to RESPONSIBLE RECEIVER.
NOTICE was prompt.
REASONABLE INSPECTION is not required-for-entitlement.
ENTITLEMENT is still-probable.

Which of the other results do you want to see?

a. The answers given to the questions
b. The unknown information
c. The assumptions which have been made
d. The conclusions which have been reached
e. All of the above
f. None of the above

: f

Do you wish to store this session? (Yes/No)
: Yes

Please enter your name or initials. (Please place a '-'
between initials and/or name.)
: TaK

Session has been saved as requested.

70-2 BCA

[¶ 8377] SOUTHWEST ENGINEERING COMPANY, INC.

ENG BCA No. 3070. June 29, 1970.

Contract No. DACW41-68-C-0177.

Changed Conditions--Subsurface Conditions--Rock Formations--Reasonable Inspection--Contractor's Knowledge.--A contractor's claim for changed conditions was rejected because certain rock formations he had discovered were not unknown physical conditions of an unusual nature. During construction of water and sewage lines, the contractor discovered rock formations that required the use of jack hammers and blasting. The contractor contended that he did not anticipate such excavation and that the government did not apprise him of the subsurface conditions. However, the contractor admitted to knowledge of some rock formations in the area gained during performance of a previous contract. In addition, it was locally known that there were rock ledges in the area. The subsurface conditions, thus, were not unknown or unusual as required by the changed conditions clause. Further, since a reasonable site inspection would have revealed the presence of rock ledges, the specifications were not deemed defective for failing to describe those conditions.

TM
[Rosie Version 2.3 11-DEC-83 17:49:48]

The name of this session shall be: SW-ENGR

You may perform any one of the following analyses:

DSC Assertion Final Payment Notice Form
Notice Promptness Responsible Receiver Government Prejudiced

Contract Obligation Excluded Conditions After-bid Conditions

Express-implied Conditions Contract Indications
Latent Deficiencies Standard Conditions
Site Inspection

Superior Knowledge-I Superior Knowledge-II
Reliance I Reliance II
Material Difference-I Material Difference-II
Exculpatory Langauge-I Exculpatory Language-II

Complete Analysis No Module

I will begin the analysis as soon as you have selected any ONE
of the options listed. Please enter the selected option
EXACTLY as it appears in the list followed by a '.'.
: Complete Analysis.

SESSION IN PROGRESS

[1] Has a differing site condition been asserted by one of
your contractors? (Yes/No)
: Yes

[2] On what date did the DSC-ASSERTION occur? (month day, year)
: December 17, 1968

[3] Has final payment been made? (Yes/No)
: No

[4] What form of notice was given to the government by
the contractor? (Select one of the following.)
a. Written notice was given.
b. Oral notice was given.
c. Other events surrounded notice.

: a

[5] On what date did the WRITTEN-NOTICE occur? (month day, year)
: November 11, 1968

[6] When was notice given in relation to the progression of work?
a. Notice given before contractor disturbed the condition.
b. Notice given before contractor performed remedial work.
c. Both a and b.
d. None of the above.
: c

[7] Is there evidence that the contractor should have been aware of the differing site condition sooner? (Yes/No)
: No

[8] To whom was notice of the differing site condition given?
a. The contracting officer
b. An authorized representative of the contracting officer
c. Any other employee of the government
d. None of the above
: a

[9] Who expressly assumed obligation for or risk of the condition in the contract?
a. The government
b. The contractor
c. Both the government and the contractor
d. Neither the government nor the contractor
: d

[10] Is the condition in question directly related to the physical conditions at the work site? (Yes/No)
: Yes

[11] Is the condition a static physical part of the work site? (Yes/No)
: Yes

[12] Did the condition occur before or after the contract award?
: Before

[13] Were there affirmatively expressed statements in the contract concerning the relevant subsurface or latent conditions at the site? (Yes/No)
: No

[14] Would the affirmatively expressed contract statements on the general conditions at the site lead a reasonable contractor to believe that the condition could be expected? (Yes/No)
: No

[15] Were indications of subsurface or latent conditions inferred from the contract as a whole? (Yes/No)
: No

There were no indications regarding the condition in the contract, therefore, a Type II condition is the most probable avenue of recovery.

[16] Was the physical data made a part of the contract? (Yes/No)
: Yes

[17] Would this physical data have indicated the nature of existing conditions to a reasonable contractor? (Yes/No)
: No

[18] By what means can the contractor establish expected conditions?
a. Customs of the trade
b. General knowledge in the industry
c. Manufacturers instructions or recommendations
d. None of the above
: b

[19] Does the actual condition differ from the standard? (Yes/No)
: No

Thinking.......

**** Entitlement will probably not be allowed under DSC claim. ****

I believe that entitlement will probably not be allowed because I
concluded that the condition does not differ from standard_of_expectation.

The following rule proved to be true causing me
to draw the above conclusion:

```
If
('expected conditions are establishable from customs-of-trade'
                              is true in answers
    or 'expected conditions are establishable from
                        general-knowledge-in-industry'
                              is true in answers
    or 'expected conditions are establishable from
                        manufacturers-recommendations'
                              is true in answers)
   and 'condition does not differ from standard-of-expectation'
                              is true in answers
   then the condition does not differ from standard_of_expectation.
```

The following statements contained by the rule
are true:

[1]
expected conditions are establishable from
general-knowledge-in-industry (in answers)

[2] condition does not differ from standard-of-expectation (in answers)

********************* END OF ANALYSIS *********************
(You should receive my bill in 7-10 days.)
(PROMPT PAYMENT IS APPRECIATED)

Do you wish to see the results from this session? (Yes/No)
: Yes

Which of the following results do you wish
to see?

a. The answers given to the questions
b. The unknown information
c. The assumptions which have been made
d. The conclusions which have been reached
e. All of the above
f. None of the above

: e

These are the results of this session:

The answers:
[ANSWERS Database]
CONDITION did occur before CONTRACT AWARD.
CONDITION does not differ from STANDARD-OF-EXPECTATION.
CONTRACTOR did not have POTENTIAL-TO-KNOW CONDITION SOONER.
GOVERNMENT did not expressly ASSUME OBLIGATION-RISK.
CONTRACTOR did not expressly ASSUME OBLIGATION-RISK.
CONTRACT did not contain STATEMENTS CONCERNING CONDITION.
CONTRACT did not contain GENERAL CONDITIONS IMPLYING CONDITION.
CONTRACT INFORMATION did contain PHYSICAL DATA.
CONTRACT did not lead ONE-TO INFER CONDITIONS.
CONTRACT INFORMATION did not reveal DEFICIENCIES/CONDITIONS.
<DECEMBER 17, 1968> is a date-of-dsc-assertion.
<NOVEMBER 11, 1968> is a date-of-written-notice.
NOTICE was received-by CONTRACTING OFFICER.
CONDITION is static PHYSICAL PART OF-WORK-SITE.
DSC was asserted.
FINAL-PAYMENT was not made.
FORM-OF-NOTICE was written.
NOTICE was given before CONTRACTOR DISTURBED CONDITION.
NOTICE was given before REMEDIAL WORK PERFORMED.
CONDITION is directly-related-to-physical-conditions.
EXPECTED CONDITIONS is establishable from GENERAL-KNOWLEDGE-IN-INDUSTRY
.

The unknowns:
[UNKNOWNS Database]

The assumptions:
[ASSUMPTIONS Database]

The conclusions:
[REPORTS Database]
CONTRACTOR did comply with WRITTEN-NOTICE-REQUIREMENT.
CONTRACTOR did comply-fully with NOTICE-REQUIREMENT.
CONDITION does not differ from STANDARD_OF_EXPECTATION.
REASON_FOR NO ENTITLEMENT does exist.
GOVERNMENT did receive NOTICE-OF-CONDITION.
CONTRACT did not assign OBLIGATION-RISK.
CONTRACT did not contain INDICATIONS CONCERNING CONDITION.
NOTICE was given-to RESPONSIBLE RECEIVER.
CONTRACTOR is not required-to INSPECT OTHER DATA.
CONTRACTOR is able-to ESTABLISH STANDARD-OF-EXPECTATION.
NOTICE was prompt.

Which of the other results do you want to see?

a. The answers given to the questions
b. The unknown information
c. The assumptions which have been made
d. The conclusions which have been reached
e. All of the above
f. None of the above

: f

Do you wish to store this session? (Yes/No)
: Yes

Please enter your name or initials. (Please place a '-' between initials and/or name.)
: TaK

Session has been saved as requested.

TM
[Rosie Version 2.3 11-DEC-83 18:05:57]

The name of this session shall be: SW-ENGR

Author of previous session: TAK

Date of previous session: 6-DEC-83

You may perform any one of the following analyses:

DSC Assertion Final Payment Notice Form
Notice Promptness Responsible Receiver Government Prejudiced

Contract Obligation Excluded Conditions After-bid Conditions

Express-implied Conditions Contract Indications
Latent Deficiencies Standard Conditions
Site Inspection

Superior Knowledge-I Superior Knowledge-II
Reliance I Reliance II
Material Difference-I Material Difference-II
Exculpatory Langauge-I Exculpatory Language-II

Complete Analysis No Module

I will begin the analysis as soon as you have selected any ONE
of the options listed. Please enter the selected option
EXACTLY as it appears in the list followed by a '.'.
: Complete Analysis.

SESSION IN PROGRESS

The differing site condition has been asserted.
The date of assertion of the differing site
condition is <DECEMBER 17, 1968>.

The final payment was not made.

The contractor did comply with the notice
requirements.

The written notice was given on <NOVEMBER 11, 1968>.

The notice was given before disturbing the condition
and before performing the work.
Therefore, notice was prompt.

The contractor should not have been aware
of the condition sooner.

Contracting officer received notice of DSC.

Neither the government nor the contractor have assumed
obligation or risk for the condition.

The condition is directly related to the physical
conditions at the site.

The condition is a static physical part of
the work site.

The condition did occur before the contract was
awarded.

There were no indications regarding the condition
in the contract, therefore, a Type II condition
is the most probable avenue of recovery.

The physical data was made part of the contract.

The physical data contained in the
contract would not have indicated the
nature of the existing condition.

The contractor can establish expected conditions
from general knowledge in the industry.

The actual condition does not differ from
the standard of expectation.

Thinking.......

**** Entitlement will probably not be allowed under DSC claim. ****

I believe that entitlement will probably not be allowed because I concluded that the condition does not differ from standard_of_expectation.

The following rule proved to be true causing me to draw the above conclusion:

```
If
('expected conditions are establishable from customs-of-trade'
                                        is true in answers
        or 'expected conditions are establishable from
                                general-knowledge-in-industry'
                                        is true in answers
        or 'expected conditions are establishable from
                                manufacturers-recommendations'
                                        is true in answers)
       and 'condition does not differ from standard-of-expectation'
                                        is true in answers
       then the condition does not differ from standard_of_expectation.
```

The following statements contained by the rule are true:

[1]
expected conditions are establishable from
general-knowledge-in-industry (in answers)

[2] condition does not differ from standard-of-expectation (in answers)

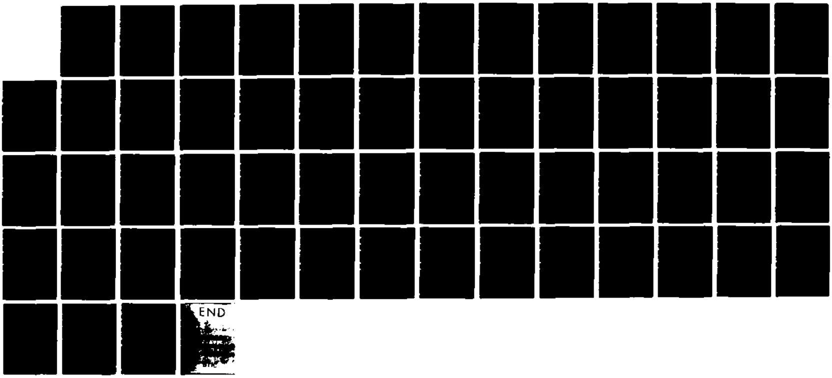
********************* END OF ANALYSIS *********************
(You should receive my bill in 7-10 days.)
(PROMPT PAYMENT IS APPRECIATED)

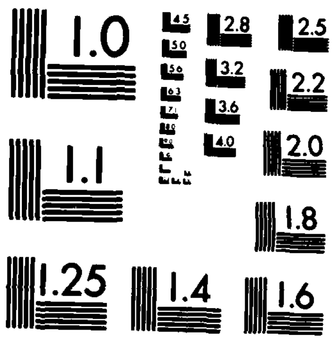
Do you wish to see the results from this session? (Yes/No)
: yYes

Which of the following results do you wish to see?

a. The answers given to the questions
b. The unknown information
c. The assumptions which have been made
d. The conclusions which have been reached
e. All of the above

AD-A146 681 THE APPLICATION OF ARTIFICIAL INTELLIGENCE TO CONTRACT 6/6
MANAGEMENT(U) CONSTRUCTION ENGINEERING RESEARCH LAB
(ARMY) CHAMPAIGN IL T A KRUPPENBACHER AUG 84
UNCLASSIFIED CERL-TM-P-166 F/G 9/1 NL





MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

f. None of the above

: e

These are the results of this session:

The answers:

[ANSWERS Database]

CONDITION did occur before CONTRACT AWARD.
CONDITION does not differ from STANDARD-OF-EXPECTATION.
CONTRACTOR did not have POTENTIAL-TO-KNOW CONDITION SOONER.
GOVERNMENT did not expressly ASSUME OBLIGATION-RISK.
CONTRACTOR did not expressly ASSUME OBLIGATION-RISK.
CONTRACT did not contain STATEMENTS CONCERNING CONDITION.
CONTRACT did not contain GENERAL CONDITIONS IMPLYING CONDITION.
CONTRACT INFORMATION did contain PHYSICAL DATA.
CONTRACT did not lead ONE-TO INFER CONDITIONS.
CONTRACT INFORMATION did not reveal DEFICIENCIES/CONDITIONS.
<DECEMBER 17, 1968> is a date-of-dsc-assertion.
<NOVEMBER 11, 1968> is a date-of-written-notice.
NOTICE was received-by CONTRACTING OFFICER.
CONDITION is static PHYSICAL PART OF-WORK-SITE.
DSC was asserted.
FINAL-PAYMENT was not made.
FORM-OF-NOTICE was written.
NOTICE was given before CONTRACTOR DISTURBED CONDITION.
NOTICE was given before REMEDIAL WORK PERFORMED.
CONDITION is directly-related-to-physical-conditions.
EXPECTED CONDITIONS is establishable from GENERAL-KNOWLEDGE-IN-INDUSTRY

.

.

The unknowns:

[UNKNOWNS Database]

The assumptions:

[ASSUMPTIONS Database]

The conclusions:

[REPORTS Database]

CONTRACTOR did comply with WRITTEN-NOTICE-REQUIREMENT.
CONTRACTOR did comply-fully with NOTICE-REQUIREMENT.
CONDITION does not differ from STANDARD_OF_EXPECTATION.
REASON_FOR NO ENTITLEMENT does exist.
GOVERNMENT did receive NOTICE-OF-CONDITION.
CONTRACT did not assign OBLIGATION-RISK.
CONTRACT did not contain INDICATIONS CONCERNING CONDITION.
CONTRACTOR is not required-to INSPECT OTHER DATA.
CONTRACTOR is able-to ESTABLISH STANDARD-OF-EXPECTATION.
NOTICE was prompt.
NOTICE was given to RESPONSIBLE RECEIVER.

Which of the other results do you want to see?

a. The answers given to the questions
b. The unknown information
c. The assumptions which have been made
d. The conclusions which have been reached
e. All of the above
f. None of the above

: f

Do you wish to store this session? (Yes/No)
: No

Session NOT saved.

TM
[Rosie Version 2.3 6-DEC-83 02:48:09]

The name of this session shall be: SWENGR2

You may perform any one of the following analyses:

DSC Assertion Final Payment Notice Form
Notice Promptness Responsible Receiver Government Prejudiced

Contract Obligation Excluded Conditions After-bid Conditions

Express-implied Conditions Contract Indications
Latent Deficiencies Standard Conditions
Site Inspection

Superior Knowledge-I Superior Knowledge-II
Reliance I Reliance II
Material Difference-I Material Difference-II
Exculpatory Langauge-I Exculpatory Language-II

Complete Analysis No Module

I will begin the analysis as soon as you have selected any ONE of the options listed. Please enter the selected option EXACTLY as it appears in the list followed by a '.'.
: Complete Analysis.

SESSION IN PROGRESS

[1] Has a differing site condition been asserted by one of your contractors? (Yes/No)
: Yes

[2] On what date did the DSC-ASSERTION occur? (month day, year)
: December 17, 1968

[3] Has final payment been made? (Yes/No)
: No

[4] What form of notice was given to the government by the contractor? (Select one of the following.)
a. Written notice was given.
b. Oral notice was given.
c. Other events surrounded notice.

: a

[5] On what date did the WRITTEN-NOTICE occur? (month day, year)
: November 11, 1968

[6] When was notice given in relation to the progression of work?
a. Notice given before contractor disturbed the condition.
b. Notice given before contractor performed remedial work.
c. Both a and b.
d. None of the above.
: c

[7] Is there evidence that the contractor should have been aware of the differing site condition sooner? (Yes/No)
: No

[8] To whom was notice of the differing site condition given?
a. The contracting officer
b. An authorized representative of the contracting officer
c. Any other employee of the government
d. None of the above
: a

[9] Who expressly assumed obligation for or risk of the condition in the contract?
a. The government
b. The contractor
c. Both the government and the contractor
d. Neither the government nor the contractor
: d

[10] Is the condition in question directly related to the physical conditions at the work site? (Yes/No)
: Yes

[11] Is the condition a static physical part of the work site? (Yes/No)
: Yes

[12] Did the condition occur before or after the contract award?
: Before

[13] Were there affirmatively expressed statements in the contract concerning the relevant subsurface or latent conditions at the site? (Yes/No)
: No

[14] Would the affirmatively expressed contract statements on the general conditions at the site lead a reasonable contractor to believe that the condition could be expected? (Yes/No)
: No

[15] Were indications of subsurface or latent conditions inferred from the contract as a whole? (Yes/No)
: No

There were no indications regarding the condition in the contract, therefore, a Type II condition is the most probable avenue of recovery.

[16] Was the physical data made a part of the contract? (Yes/No)
: Yes

[17] Would this physical data have indicated the nature of existing conditions to a reasonable contractor? (Yes/No)
: No

[18] By what means can the contractor establish expected conditions?
a. Customs of the trade
b. General knowledge in the industry
c. Manufacturers instructions or recommendations
d. None of the above
: -

[19] Did the GOVERNMENT know about the ACTUAL-CONDITIONS? (Yes/No)
: Yes

[20] Did the government reveal ACTUAL-CONDITIONS to the contractor? (Yes/No)
: No

Oversight, misrepresentation or concealment on
the part of the government. A possible
breach has occurred. However, I will
continue with the analysis.

[21] Was the contractor in possession of unilateral superior
knowledge of the actual conditions? (Yes/No)
: No

[22] Would simple inquiries by the contractor have
revealed the condition to be contrary to the
STANDARD-OF-EXPECTATION? (Yes/No)
: Yes

It has become obvious that the contractor did
not make simple inquiries.

Thinking.......

**** Entitlement will probably not be allowed under DSC claim. ****

I believe that entitlement will probably not be allowed because I
concluded that the contractor did not make simple inquiries.

The following rule proved to be true causing me
to draw the above conclusion:

If
'simple inquiry did give potential to-know contrary conditions'
is true in answers
then the contractor did not make simple inquiries.

The following statements contained by the rule
are true:

[1] simple inquiry did give potential to-know contrary conditions (in answers
)

********************* END OF ANALYSIS *********************
(You should receive my bill in 7-10 days.)
(PROMPT PAYMENT IS APPRECIATED)

Do you wish to see the results from this session? (Yes/No)
: Yes

Which of the following results do you wish
to see?

a. The answers given to the questions
b. The unknown information
c. The assumptions which have been made
d. The conclusions which have been reached
e. All of the above
f. None of the above

: e

These are the results of this session:

The answers:
[ANSWERS Database]
CONDITION did occur before CONTRACT AWARD.
CONTRACTOR did not have POTENTIAL-TO-KNOW CONDITION SOONER.
GOVERNMENT did not expressly ASSUME OBLIGATION-RISK.
CONTRACTOR did not expressly ASSUME OBLIGATION-RISK.
CONTRACT did not contain STATEMENTS CONCERNING CONDITION.
CONTRACT did not contain GENERAL CONDITIONS IMPLYING CONDITION.
CONTRACT INFORMATION did contain PHYSICAL DATA.
CONTRACT did not lead ONE-TO INFER CONDITIONS.
CONTRACT INFORMATION did not reveal DEFICIENCIES/CONDITIONS.
GOVERNMENT did not reveal ACTUAL-CONDITIONS.
SIMPLE INQUIRY did give POTENTIAL TO-KNOW CONTRARY CONDITIONS.
<DECEMBER 17, 1968> is a date-of-dsc-assertion.
<NOVEMBER 11, 1968> is a date-of-written-notice.
NOTICE was received-by CONTRACTING OFFICER.
GOVERNMENT was aware-of ACTUAL-CONDITIONS.
CONTRACTOR was not aware-of ACTUAL-CONDITIONS.
CONDITION is static PHYSICAL PART OF-WORK-SITE.
DSC was asserted.
FINAL-PAYMENT was not made.
FORM-OF-NOTICE was written.
NOTICE was given before CONTRACTOR DISTURBED CONDITION.
NOTICE was given before REMEDIAL WORK PERFORMED.
CONDITION is directly-related-to-physical-conditions.

The unknowns:
[UNKNOWNS Database]

The assumptions:
[ASSUMPTIONS Database]

The conclusions:
[REPORTS Database]

CONTRACTOR did comply with WRITTEN-NOTICE-REQUIREMENT.
CONTRACTOR did comply-fully with NOTICE-REQUIREMENT.
REASON_FOR NO ENTITLEMENT does exist.
GOVERNMENT did receive NOTICE-OF-CONDITION.
CONTRACT did not assign OBLIGATION-RISK.
CONTRACT did not contain INDICATIONS CONCERNING CONDITION.
GOVERNMENT did have SUPERIOR KNOWLEDGE.
CONTRACTOR did not make SIMPLE INQUIRIES.
NOTICE was given-to RESPONSIBLE RECEIVER.
CONTRACTOR is not required-to INSPECT OTHER DATA.
NOTICE was prompt.

Which of the other results do you want to see?

a. The answers given to the questions
b. The unknown information
c. The assumptions which have been made
d. The conclusions which have been reached
e. All of the above
f. None of the above

: f

Do you wish to store this session? (Yes/No)
: Yes

Please enter your name or initials. (Please place a '-'
between initials and/or name.)
: TaK

Session has been saved as requested.

TM
[Rosie Version 2.3 6-DEC-83 04:31:20]

The name of this session shall be: SWENGR2

Author of previous session: TAK

Date of previous session: 6-DEC-83

You may perform any one of the following analyses:

DSC Assertion Final Payment Notice Form
Notice Promptness Responsible Receiver Government Prejudiced

Contract Obligation Excluded Conditions After-bid Conditions

Express-implied Conditions Contract Indications
Latent Deficiencies Standard Conditions
Site Inspection

Superior Knowledge-I Superior Knowledge-II
Reliance I Reliance II
Material Difference-I Material Difference-II
Exculpatory Langauge-I Exculpatory Language-II

Complete Analysis No Module

I will begin the analysis as soon as you have selected any ONE
of the options listed. Please enter the selected option
EXACTLY as it appears in the list followed by a '.'.
: Complete Analysis.

SESSION IN PROGRESS

The differing site condition has been asserted.
The date of assertion of the differing site
condition is <DECEMBER 17, 1968>.

The final payment was not made.

The contractor did comply with the notice
requirements.

The written notice was given on <NOVEMBER 11, 1968>.

The notice was given before disturbing the condition
and before performing the work.
Therefore, notice was prompt.

The contractor should not have been aware
of the condition sooner.

Contracting officer received notice of DSC.

Neither the government nor the contractor have assumed
obligation or risk for the condition.

The condition is directly related to the physical
conditions at the site.

The condition is a static physical part of
the work site.

The condition did occur before the contract was
awarded.

There were no indications regarding the condition
in the contract, therefore, a Type II condition
is the most probable avenue of recovery.

The physical data was made part of the contract.

The physical data contained in the
contract would not have indicated the
nature of .he existing condition.

When we last spoke you were unsure as to the means
by which the contractor could establish expected
conditions.
Have you discovered an answer to this question yet? (Yes/No)
: No

The government was aware of the ACTUAL-CONDITIONS.

The government did not communicate its knowledge
of the ACTUAL-CONDITIONS to the contractor.
Possible breach by misrepresentation has occurred
however, I will continue our analysis.

The contractor was not aware of the ACTUAL-CONDITIONS.

It has become obvious that the contractor did
not simple inquiries.
These simple inquiries by the contractor would
have revealed the condition.

Thinking.......

**** Entitlement will probably not be allowed under DSC claim. ****

I believe that entitlement will probably not be allowed because I
concluded that the contractor did not make simple inquiries.

The following rule proved to be true causing me
to draw the above conclusion:

If
'simple inquiry did give potential to-know contrary conditions'
is true in answers
then the contractor did not make simple inquiries.

The following statements contained by the rule
are true:

[1] simple inquiry did give potential to-know contrary conditions (in answers
)

********************* END OF ANALYSIS *********************
(You should receive my bill in 7-10 days.)
(PROMPT PAYMENT IS APPRECIATED)

Do you wish to see the results from this session? (Yes/No)
: Yes

Which of the following results do you wish
to see?

a. The answers given to the questions
b. The unknown information
c. The assumptions which have been made
d. The conclusions which have been reached
e. All of the above
f. None of the above

: e

These are the results of this session:

The answers:
[ANSWERS Database]
CONDITION did occur before CONTRACT AWARD.
CONTRACTOR did not have POTENTIAL-TO-KNOW CONDITION SOONER.
GOVERNMENT did not expressly ASSUME OBLIGATION-RISK.
CONTRACTOR did not expressly ASSUME OBLIGATION-RISK.
CONTRACT did not contain STATEMENTS CONCERNING CONDITION.
CONTRACT did not contain GENERAL CONDITIONS IMPLYING CONDITION.
CONTRACT INFORMATION did contain PHYSICAL DATA.
CONTRACT did not lead ONE-TO INFER CONDITIONS.
CONTRACT INFORMATION did not reveal DEFICIENCIES/CONDITIONS.
GOVERNMENT did not reveal ACTUAL-CONDITIONS.
SIMPLE INQUIRY did give POTENTIAL TO-KNOW CONTRARY CONDITIONS.
<DECEMBER 17, 1968> is a date-of-dsc-assertion.
<NOVEMBER 11, 1968> is a date-of-written-notice.
NOTICE was received-by CONTRACTING OFFICER.
GOVERNMENT was aware-of ACTUAL-CONDITIONS.
CONTRACTOR was not aware-of ACTUAL-CONDITIONS.
CONDITION is static PHYSICAL PART OF-WORK-SITE.
DSC was asserted.
FINAL-PAYMENT was not made.
FORM-OF-NOTICE was written.
NOTICE was given before CONTRACTOR DISTURBED CONDITION.
NOTICE was given before REMEDIAL WORK PERFORMED.
CONDITION is directly-related-to-physical-conditions.

The unknowns:
[UNKNOWNS Database]
IF-EXPECTED CONDITIONS is establishable.

The assumptions:
[ASSUMPTIONS Database]
CONTRACTOR is able-to ESTABLISH STANDARD-OF-EXPECTATION.

The conclusions:
[REPORTS Database]
CONTRACTOR did comply with WRITTEN-NOTICE-REQUIREMENT.

CONTRACTOR did comply-fully with NOTICE-REQUIREMENT.
REASON_FOR NO ENTITLEMENT does exist.
GOVERNMENT did receive NOTICE-OF-CONDITION.
CONTRACT did not assign OBLIGATION-RISK.
CONTRACT did not contain INDICATIONS CONCERNING CONDITION.
GOVERNMENT did have SUPERIOR KNOWLEDGE.
CONTRACTOR did not make SIMPLE INQUIRIES.
CONTRACTOR is not required-to INSPECT OTHER DATA.
NOTICE was prompt.
NOTICE was given to RESPONSIBLE RECEIVER.

Which of the other results do you want to see?

a. The answers given to the questions
b. The unknown information
c. The assumptions which have been made
d. The conclusions which have been reached
e. All of the above
f. None of the above

: f

Do you wish to store this session? (Yes/No)
: No

Session NOT saved.

77-1 BCA

[¶ 12,228] TECTONICS, INC.

VACAB No. 1187. November 16, 1976.

Contract No. V546C-66.

Specifications--Defective Specifications and Impossibility of Performance--Drawing Defective.

An electrical subcontractor was not entitled to additional costs allegedly incurred as a result of faulty drawings because the work to be performed was described with sufficient clarity to have enabled the subcontractor to understand the government's requirements. The subcontractor claimed the alleged error pictured the main line he was to install as being approximately 285 feet in length, when in fact his completed distance was 498 feet. However, the drawing he alleged misled him was intended only as a guide to describe what was to be done. It did not direct that the line be laid in any specific direction, and the route was left to the discretion of the subcontractor. Thus, the drawings were adequate for their intended purpose. The subcontractor should also not have been misled by the drawings since he had the option to review the government's requirements and conduct a proper on-site inspection.

TM
[Rosie Version 2.3 3-DEC-83 22:52:05]

The name of this session shall be: TECTON

You may perform any one of the following analyses:

DSC Assertion   Final Payment   Notice Form
Notice Promptness   Responsible Receiver   Government Prejudiced

Contract Obligation   Excluded Conditions   After-bid Conditions

Express-implied Conditions   Contract Indications
Latent Deficiencies   Standard Conditions
Site Inspection

Superior Knowledge-I   Superior Knowledge-II
Reliance I   Reliance II
Material Difference-I   Material Difference-II
Exculpatory Langauge-I   Exculpatory Language-II

Complete Analysis   No Module

I will begin the analysis as soon as you have selected any ONE of the options listed. Please enter the selected option EXACTLY as it appears in the list followed by a '.'.
: Complete Analysis.

SESSION IN PROGRESS

[1] Has a differing site condition been asserted by one of your contractors? (Yes/No)
: Yes

[2] On what date did the DSC-ASSERTION occur? (month day, year)
: September 2, 1975

[3] Has final payment been made? (Yes/No)
: No

[4] What form of notice was given to the government by the contractor? (Select one of the following.)

a. Written notice was given.
b. Oral notice was given.
c. Other events surrounded notice.

: a

[5] On what date did the WRITTEN-NOTICE occur? (month day, year)
: September 2, 1975

[6] When was notice given in relation to the progression of work?
a. Notice given before contractor disturbed the condition.
b. Notice given before contractor performed remedial work.
c. Both a and b.
d. None of the above.
: d

[7] Is there evidence that the contractor should have been aware of the differing site condition sooner? (Yes/No)
: NYes

[8] To whom was notice of the differing site condition given?
a. The contracting officer
b. An authorized representative of the contracting officer
c. Any other employee of the government
d. None of the above
: -a

[9] Is the government able to show that evidence against the claim was obscurred due to the passage of time? (Yes/No)
: No

[10] Is the government able to show that proper notice would have resulted in cheaper resolution of the condition by CO? (Yes/No)
: Yes

[11] Can the government show that it suffered any other prejudice through inadequacy of or lack of notice? (Yes/No)
: No

The government did suffer prejudice.

[12] Was the govenment defense made impossible by inadequacy of or lack of notice? (Yes/No)
: No

[13] Was the contractor able to provide sufficient additional proof of claim sufficient to prove entitlement to DSC claim in spite of slight prejudice to the government? (Yes/No)
: No

Inadequacy of or lack of notice plus resulting prejudice to the government render entitlement to DSC claim unlikely. However, I will continue our analysis.

[14] Who expressly assumed obligation for or risk of the condition in the contract?
a. The government
b. The contractor
c. Both the government and the contractor
d. Neither the government nor the contractor
: d

[15] Is the condition in question directly related to the physical conditions at the work site? (Yes/No)
: Yes

[16] Is the condition a static physical part of the work site? (Yes/No)
: Yes

[17] Did the condition occur before or after the contract award?
: Before

[18] Were there affirmatively expressed statements in the contract concerning the relevant subsurface or latent conditions at the site? (Yes/No)
: Yes

There were indications regarding the condition in the contract, therefore, a Type I condition is the most probable avenue of recovery.

[19] Were there latent deficiencies in the contract indications? (Yes/No)
: -

[20] Was the physical data made a part of the contract? (Yes/No)
: No

[21] Did the GOVERNMENT know about the CONTRACT-DEFICIENCY? (Yes/No)
: -

[22] Was the contractor in possession of unilateral superior or imputed knowledge that the actual conditions patently differed from the indications in the contract? (Yes/No)
: No

[23] Would simple inquiries by the contractor have revealed the condition to be contrary to the CONTRACT-INDICATIONS? (Yes/No)
: -

[24] Did any of the following acts of the government hinder the site inspection?

a. Access to site was denied
b. Inadequate time was allowed
c. No act of the government hindered inspection
: c

[25] Did the contractor conduct a site inspection? (Yes/No)
: Yes

[26] Was the inspection conducted by the contractor a reasonable inspection, i.e., one comparable to that expected from a reasonable, prudent contractor experienced in that particular field of work? (Yes/No)
: No

[27] Would the actual condition have been discernable

by a layman contractor performing a reasonable
inspection? (Yes/No)
: Yes

Actual conditions would have been discernable
from a reasonable inspection.

Thinking.......

**** Entitlement will probably not be allowed under DSC claim. ****

I believe that entitlement will probably not be allowed because I
concluded that the reasonable inspection is required-for-entitlement.

The following rule proved to be true causing me
to draw the above conclusion:

If
'reasonable inspection did have potential-to-reveal condition'
is true in answers
then the reasonable inspection is required-for-entitlement.

The following statements contained by the rule
are true:

[1] reasonable inspection did have potential-to-reveal condition (in answers)

********************* END OF ANALYSIS *********************
(You should receive my bill in 7-10 days.)
(PROMPT PAYMENT IS APPRECIATED)

Do you wish to see the results from this session? (Yes/No)
: |Yes

Which of the following results do you wish
to see?

a. The answers given to the questions
b. The unknown information
c. The assumptions which have been made

d. The conclusions which have been reached
e. All of the above
f. None of the above
: e

These are the results of this session:

The answers:
[ANSWERS Database]
CONDITION did occur before CONTRACT AWARD.
CONTRACTOR did have POTENTIAL-TO-KNOW CONDITION SOONER.
REASONABLE INSPECTION did have POTENTIAL-TO-REVEAL CONDITION.
PASSAGE-OF-TIME did not obscure EVIDENCE.
IMPROPER-NOTICE did cause ADDITIONAL COST.
NOTHING-ELSE did cause PREJUDICE.
GOVERNMENT did not expressly ASSUME OBLIGATION-RISK.
CONTRACTOR did not expressly ASSUME OBLIGATION-RISK.
CONTRACT did contain STATEMENTS CONCERNING CONDITION.
CONTRACT INFORMATION did not contain PHYSICAL DATA.
GOVERNMENT did not hinder INSPECTION.
CONTRACTOR did conduct SITE-INSPECTION.
ADDITIONAL-PROOF does not exist TO-PROVE-ENTITLEMENT.
<SEPTEMBER 2, 1975> is a date-of-dsc-assertion.
<SEPTEMBER 2, 1975> is a date-of-written-notice.
NOTICE was received-by CONTRACTING OFFICER.
DEFENSE-AGAINST CLAIM was not made IMPOSSIBLE.
CONTRACTOR was not aware-of CONTRACT-DEFICIENCY.
SITE-ISPECTION was not reasonable INSPECTION.
CONDITION is static PHYSICAL PART OF-WORK-SITE.
DSC was asserted.
FINAL-PAYMENT was not made.
FORM-OF-NOTICE was written.
NOTICE was not given before REMEDIAL WORK PERFORMED.
CONDITION is directly-related-to-physical-conditions.

The unknowns:
[UNKNOWNS Database]
IF-CONTRACT INDICATIONS did contain LATENT DEFICIENCIES.
IF-SIMPLE INQUIRY did give POTENTIAL TO-KNOW CONTRARY CONDITIONS.
IF-GOVERNMENT was aware-of CONTRACT-DEFICIENCY.

The assumptions:
[ASSUMPTIONS Database]
CONTRACT INDICATIONS did contain LATENT DEFICIENCIES.
SIMPLE INQUIRY did not give POTENTIAL TO-KNOW CONTRARY CONDITIONS.
GOVERNMENT was not aware-of CONTRACT-DEFICIENCY.

The conclusions:
[REPORTS Database]
CONTRACTOR did comply with WRITTEN-NOTICE-REQUIREMENT.

REASON_FOR NO ENTITLEMENT does exist.
GOVERNMENT did receive NOTICE-OF-CONDITION.
ADDITIONAL-COST did cause PREJUDICE.
GOVERNMENT did suffer PREJUDICE.
CONTRACT did not assign OBLIGATION-RISK.
CONTRACT did contain INDICATIONS CONCERNING CONDITION.
PROMPTNESS_OF_NOTICE is an in_doubt.
NOTICE was given-to RESPONSIBLE RECEIVER.
REASONABLE INSPECTION is required-for-entitlement.

Which of the other results do you want to see?

a. The answers given to the questions
b. The unknown information
c. The assumptions which have been made
d. The conclusions which have been reached
e. All of the above
f. None of the above

: f

Do you wish to store this session? (Yes/No)
: No

Session NOT saved.

77-1 BCA

[¶ 12,322] WELCH CONSTRUCTION COMPANY, INC.

PSBCA No. 217. February 11, 1977.

Contract No. 059973-75-R-0106.

Changed Conditions--Differing Site Conditions Clause--Rain.

A construction contractor was not entitled to an equitable adjustment pursuant to the Differing Site Conditions clause for delay and extra work occasioned by the interaction of abnormal, unseasonal rains with the impervious soil at the worksite because the impervious nature of the soil had been disclosed to the contractor and severe weather must interact with a misrepresented, unknown, or unusual subsurface condition for a contractor to recover under the Differing Site Conditions clause. The contractor contended that the unusually heavy rainfall, a layer of clay overlaying the hardpan (a highly compressed layer of clayey soil), and pockets of free water in the hardpan adjacent the building areas constituted differing site conditions. Severe weather itself is not a risk which is covered by the Differing Site Conditions clause, unless it interacts with an undisclosed subsurface condition. The hardpan, the primary cause of the wet soil problem, was disclosed to the contractor in a soil report. The thin layer of clay had not been disclosed because it was at a depth to which excavation would not have extended but for the

supersaturation of the soil. However, the clay did not contribute to the retention of moisture in the soil above the hardpan.

TM
[Rosie Version 2.3 5-DEC-83 00:37:11]

The name of this session shall be: WELCH

You may perform any one of the following analyses:

DSC Assertion Final Payment Notice Form
Notice Promptness Responsible Receiver Government Prejudiced

Contract Obligation Excluded Conditions After-bid Conditions

Express-implied Conditions Contract Indications
Latent Deficiencies Standard Conditions
Site Inspection

Superior Knowledge-I Superior Knowledge-II
Reliance I Reliance II
Material Difference-I Material Difference-II
Exculpatory Langauge-I Exculpatory Language-II

Complete Analysis No Module

I will begin the analysis as soon as you have selected any ONE of the options listed. Please enter the selected option EXACTLY as it appears in the list followed by a '.'.
: Complete Ansalysis.

SESSION IN PROGRESS

[1] Has a differing site condition been asserted by one of your contractors? (Yes/No)
: Yes

[2] On what date did the DSC-ASSERTION occur? (month day, year)
: April 15, 1975

[3] Has final payment been made? (Yes/No)
: No

[4] What form of notice was given to the government by the contractor? (Select one of the following.)

a. Written notice was given.
b. Oral notice was given.
c. Other events surrounded notice.

: a

[5] On what date did the WRITTEN-NOTICE occur? (month day, year)
: April 7, 1975

[6] When was notice given in relation to the progression of work?
a. Notice given before contractor disturbed the condition.
b. Notice given before contractor performed remedial work.
c. Both a and b.
d. None of the above.
: c

[7] Is there evidence that the contractor should have been aware of the differing site condition sooner? (Yes/No)
: -

[8] To whom was notice of the differing site condition given?
a. The contracting officer
b. An authorized representative of the contracting officer
c. Any other employee of the government
d. None of the above
: a

[9] Who expressly assumed obligation for or risk of the condition in the contract?
a. The government
b. The contractor
c. Both the government and the contractor
d. Neither the government nor the contractor
: d

[10] Is the condition in question directly related to the physical conditions at the work site? (Yes/No)
: Yes

[11] Is the condition a static physical part of the work site? (Yes/No)
: Yes

[12] Did the condition occur before or after the contract award?
: After

Differing site condition is usually limited to
those conditions occurring before bid or contract award.
However, an exception may be allowed.

[13] What or who did the differing site condition result
from? (If you are unfamiliar with this question type '?'
then hit return)
: Act-of-God

: end.

[14] Was the condition caused by an interaction of a
non-compensable condition with physical factors
at the site? (Yes/No)
: Yes

Since an interaction with the physical factors
at the site did occur, the condition may be an
exception to the after-bid exclusions and may be
within the scope of the clause.

[15] Were there affirmatively expressed statements in the
contract concerning the relevant subsurface or
latent conditions at the site? (Yes/No)
: Yes

There were indications regarding the condition
in the contract, therefore, a Type I condition is
the most probable avenue of recovery.

[16] Were there latent deficiencies in the contract
indications? (Yes/No)
: -

[17] Was the physical data made a part of the
contract? (Yes/No)
: Yes

[18] Would this physical data have revealed or resolved
the apparent deficiencies in the contract indications for

a reasonable contractor? (Yes/No)
: Yes

Thinking.......

**** Entitlement will probably not be allowed under DSC claim. ****

I believe that entitlement will probably not be allowed because I concluded that the contractor did fail to-heed contract indications.

The following rule proved to be true causing me to draw the above conclusion:

If
'contract indications did not contain latent deficiencies'
is true in answers
or ('contract information did contain physical data'
is true in answers
and 'contract information did reveal deficiencies/conditions'
is true in answers)
then the contractor did fail to-heed contract indications.

The following statements contained by the rule are true:

[1] contract information did contain physical data (in answers)

[2] contract information did reveal deficiencies/conditions (in answers)

********************* END OF ANALYSIS *********************
(You should receive my bill in 7-10 days.)
(PROMPT PAYMENT IS APPRECIATED)

Do you wish to see the results from this session? (Yes/No)
: Yes

Which of the following results do you wish to see?

a. The answers given to the questions
b. The unknown information
c. The assumptions which have been made
d. The conclusions which have been reached

e. All of the above
f. None of the above
: e

These are the results of this session:

The answers:
[ANSWERS Database]
CONDITION did occur after CONTRACT AWARD.
CONDITION did result from ACT-OF-GOD.
CONDITION did result from
INTERACTING-NON-COMPENSABLE-AND-PHYSICAL-FACTORS.
GOVERNMENT did not expressly ASSUME OBLIGATION-RISK.
CONTRACTOR did not expressly ASSUME OBLIGATION-RISK.
CONTRACT did contain STATEMENTS CONCERNING CONDITION.
CONTRACT INFORMATION did contain PHYSICAL DATA.
CONTRACT INFORMATION did reveal DEFICIENCIES/CONDITIONS.
<APRIL 15, 1975> is a date-of-dsc-assertion.
<APRIL 7, 1975> is a date-of-written-notice.
NOTICE was received-by CONTRACTING OFFICER.
CONDITION is static PHYSICAL PART OF-WORK-SITE.
DSC was asserted.
FINAL-PAYMENT was not made.
FORM-OF-NOTICE was written.
NOTICE was given before CONTRACTOR DISTURBED CONDITION.
NOTICE was given before REMEDIAL WORK PERFORMED.
CONDITION is directly-related-to-physical-conditions.

The unknowns:
[UNKNOWNS Database]
IF-CONTRACTOR did have POTENTIAL-TO-KNOW CONDITION SOONER.
IF-CONTRACT INDICATIONS did contain LATENT DEFICIENCIES.

The assumptions:
[ASSUMPTIONS Database]
CONTRACTOR did not have POTENTIAL-TO-KNOW CONDITION SOONER.
CONTRACT INDICATIONS did contain LATENT DEFICIENCIES.

The conclusions:
[REPORTS Database]
CONTRACTOR did comply with WRITTEN-NOTICE-REQUIREMENT.
CONTRACTOR did comply-fully with NOTICE-REQUIREMENT.
REASON_FOR NO ENTITLEMENT does exist.
GOVERNMENT did receive NOTICE-OF-CONDITION.
CONTRACT did not assign OBLIGATION-RISK.
CONTRACT did contain INDICATIONS CONCERNING CONDITION.
CONTRACTOR did fail TO-HEED CONTRACT INDICATIONS.
NOTICE was given-to RESPONSIBLE RECEIVER.
CONDITION is possible EXCEPTION-TO AFTER-BID EXCLUSIONS.
NOTICE was prompt.

Which of the other results do you want to see?

a. The answers given to the questions
b. The unknown information
c. The assumptions which have been made
d. The conclusions which have been reached
e. All of the above
f. None of the above

: f

Do you wish to store this session? (Yes/No)
: Yes

Please enter your name or initials. (Please place a '-' between initials and/or name.)
: TaK

Session has been saved as requested.

APPENDIX D

PARTIAL LISTING OF DSC CASES

Arundel Corp. v. United States	103 Ct. Cl. 688 326 U.S. 752 (1945)
Tacome Dredging Co. v. U.S.	52 Ct. Cls. 447
Turnkey Enterprises, Inc., v. U.S.	220 Ct. Cl. 597 F.2d 750 (1979)
John McShain, Inc. v. U.S.	179 Ct. Cl. 632 375 F.2d 829 (1967)
Overland Elect. Co.	ASBCA 9096 1964 BCA
E. W. Jackson Contr. Co.	ASBCA 7267 1962 BCA
George A. Fuller Co.	ASBCA 8524 1962 BCA
Lenry, Inc., v. U.S.	156 Ct. Cl. 46 297 F.2d 550 (1962)
Hardeman-Monier-Hutcherson v. U.S.	198 Ct. Cl. 472 458 F.2d 1364 (1972)
National Bank of Kansas City v. U.S.	184 Ct. Cl. 741 397 F.2d 984 (1968)
Warren Painting Co.	ASBCA 18456 74-2 BCA (1974)
Jim Challinor	AGBCA 75-133 78-2 BCA
Roen Salvage Co.	ENGBCA 3670 79-2 BCA (1979)
Cf. Bosen & Dybevik Constr. Co.	AGBCA 243 69-1 BCA (1969)
Mittry v. U.S.	73 Ct. Cl. 341 (1931)
Lee Hoffman v. U.S.	166 Ct. Cl. 39 340 F.2d 645 (1964)
Frank W. Miller Constr. Co.	ASBCA 22347 78-1 BCA (1978)

Harding Equip't Co.	ASBCA 2477 6 CCF (1955)
Concrete Constr. Corp.	IBCA 432-64 65-1 BCA (1964)
Acme Missiles & Constr. Corp.	ASBCA 10784 66-1 BCA (1966)
Dave & Gerben Contracting Co.	ASBCA 6257 1962 BCA (1974)
F. D. Rich Co.	ASBCA 6515 1963 BCA
Welch Construction Co.	PSBCA 217 77-1 BCA (1977)
Monmouth Fund, Inc.	ASBCA 20158 77-1 BCA (1976)
John A. Johnson Contr. Corp. v. U.S.	143 Ct. Cl. 645 132 F. Supp. 698 (1955)
Premier Electrical Constr. Co.	FAACAP 66-10 65-2 BCA (1965)
Bateson Stolte, Inc. v. U.S.	145 Ct. Cl. 387 172 F. Supp. 454 (1959)
Koppers Co., Malan Constr. Dept.	ENGBCA 2699 67-2 BCA (1967)
Robert E. McKee, Inc.	ASBCA 5621 60-1 BCA (1960)
Hallman v. U.S.	112 Ct. Cl. 170 80 F. Supp. 370 (1948)
Central Florida Constr. Co.	IBCA 246 61-1 BCA (1961)
Keang Nam Enterprises Ltd.	ASBCA 13747 69-1 BCA (1969)
Charney Constr. Corp.	FAACAP 67-2 66-1 BCA (1966)
Cross Construction Co.	ENGBCA 3676 79-1 BCA (1979)

Ames Denning Inc.	ASBCA 6956 1962 BCA
George A. Rutherford Co.	NASABCA 12 1962 BCA
Yarno & Associates	ASBCA 10257 67-1 BCA (1967)
Grenco Services, Inc.	NASABCA 867-27 69-2 BCA (1969)
Lee R. Smith	ASBCA 11135 66-2 BCA (1968)
R. A. Heintz Constr. Co.	ENGBCA 3380 74-1 BCA (1974)
Freeman Electric Constr. Co.	DOTCAB 74-23 77-1 BCA (1976)
Dunbar & Sullivan Dredging Co.	ENGBCA 3165 73-2 BCA (1973)
Rottau Electric Co.	ASBCA 20283 76-BCA (1976)
Lockheed Shipbuilding & Constr. Co.	ENGBCA 3141 74-1 BCA (1974)
F. D. Rich Co.	ASBCA 14023 75-2 BCA (1975)
Okland Constr. Co.	GSBCA 3097 72-1 BCA (1972)
Penn York Constr. Corp.	ASBCA 11419 70-1 BCA (1970)
Norair Engineering Corp.	ENGBCA 3568 77-1 BCA (1976)
Foster Constr. C. A. v. U.S.	193 Ct. Cl. 587 435 F.2d 873 (1970)
Stock & Grove, Inc., v. U.S.	204 Ct. Cl. 103 493 F.2d 629 (1974)
Robertson C. v. U.S.	194 Ct. Cl. 289 437 F.2d 1360 (1971)

J. F. Whalen	ENGBCA 2859 69-1 BCA (1969)
Mann Constr. Co.	AGBCA 76-109 80-2 BCA (1980)
Aquirre Associates	AGBCA 78-120 80-2 BCA (1980)
Pacific Alaska Constr., Inc. v. U.S.	193 Ct. Cl. 850 436 F.2d 461 (1971)
Framlau Corp.	ASBCA 14205 71-2 BCA (1971)
Smith Constr. Co. v. U.S.	188 Ct. Cl. 1062 412 F.2d 1325 (1969)
Fred Benvenuti, Inc.	DOTCAB 74-25 75-2 BCA (1975)
Corner Construction Co.	ASBCA 20156 75-1 BCA (1975)
Dravo Corp.	ENGBCA 3901 80-2 BCA (1980)
Charles T. Parker Constr. Co. v. U.S.	193 Ct. Cl. 320 433 F.2d 771 (1970)
Husman Bros., Inc.	DOTCAB 71-15 73-1 BCA (1973)
Western Well Drilling Co., v. U.S.	96 F. Supp. 377 (D. Calif. 1951)
Fairbanks Builders, Inc.	ASBCA 18288 74-2 BCA (1974)
John C. Grimberg Co.	ASBCA 15218 73-1 BCA (1972)
Layne Texas Co.	IBCA 362 65-1 BCA (1965)
Redman Service, Inc.	ASBCA 8853 1963 BCA
Daymar, Inc.	DOTCAB 77-13 78-1 BCA (1977)

Community Power Suction Furnace Cl. Co.	ASBCA 13803 69-2 BCA (1969)
Transco Contracting Co.	VACAB 921 71-2 BCA (1971)
George E. Jensen, Inc.	ASBCA 20234 76-1 BCA (1976)
Edgar M. Williams	ASBCA 16058 72-2 BCA (1972)
Kemmons-Wilson, Inc.	ASBCA 16167 72-2 BCA (1972)
Leal v. U.S.	149 Ct. Cl. 451 276 F.2d 378 (1960)
Ragonese v. U.S.	128 Ct. Cl. 156 120 F. Supp. 768 (1954)
J. A. Jones Constr. Co. v. U.S.	182 Ct. Cl. 615 390 F.2d 806 (1968)
Maryland Painting Co.	ENGBCA 3337 73-2 BCA (1973)
Leonard Blinderman Constr. Co.	ASBCA 18946 75-1 BCA (1975)
Alps Constr. Corp.	ASBCA 1966 73-2 BCA (1973)
Mojave Enterprises	AGBCA 75-114 77-1 BCA (1977)
Bernard McMenamy Contractor, Inc.	ENGBCA 3414 77-1 BCA (1976)
Continental Drilling Co.	ENGBCA 3455 75-2 BCA (1975)
Kaiser Industries v. U.S.	169 Ct. Cl. 310 340 F.2d 322 (1965)
Pavement Specialists, Inc.	ASBCA 17410 73-2 BCA (1973)
Raymond International, Inc.	ASBCA 13121 70-1 BCA (1970)

Cal-Pacific Foresters	AGBCA 230 70-1 BCA (1969)
E. Arthur Higgins	AGBCA 76-128 79-2 BCA (1979)
Cottrell Engineering Corp.	ENGBCA 3964 80-1 BCA (1979)
A. D. Herman Constr. Co.	GSBCA 4823 78-1 BCA (1978)
E. R. McKee v. U.S.	205 Ct. Cl. 303 500 F.2d 525 (1974)
Quiller Constr. Co.	ASBCA 8053 1963 BCA
S. T. G. Constr. Co. v. U.S.	157 Ct. Cl. 409 (1962)
Reid Contracting Co.	IBCA 74 58-2 BCA (1958)
Vann v. U.S.	190 Ct. Cl. 546 420 F.2d 968 (1970)
Tectonics, Inc.	VACAB 1187 77-1 BCA (1976)
Ray D. Bolander Co.	IBCA 331 65-2 BCA (1965)
Diversacon Industries, Inc.	ENGBCA 3365 75-1 BCA (1975)
Charles T. Parker Constr. Co.	DCAB PR-41 65-1 BCA (1965)
T & B. Builders, Inc.	ENGBCA 3664 77-2 BCA (1977)
J. J. Welcome Constr. Co.	ASBCA 19653 75-1 BCA (1974)
Piracci Constr. Co.	GSBCA 2793 70-1 BCA (1970)
S. Kane & Sons, Inc.	VACAB 1254 78-1 BCA (1978)

M. M. Sundt Constr. Co.	ASBCA 17475 74-1 BCA (1974)
Shepherd v. U.S.	125 Ct. Cl. 724 113 F. Supp. 648 (1953)
McClosky & Co.	PSBCA 497 74-1 BCA (1974)
R. C. Hedreen Co.	GSBCA 42 89 77-1 BCA (1977)
William F. Klingensmith	GSBCA 3161 71-2 BCA (1971)
Peterson Sharpe Engineering Corp.	ASBCA 18780 77-1 BCA (1977)
C. H. Leavell & Co.	ASBCA 16099 72-2 BCA (1972)
Parcoa, Inc.	AGBCA 76-130 77-2 BCA (1977)
R. R. Tyler	AGBCA 381 77-1 BCA (1977)
De Mauro Constr. Co.	ASBCA 17029 77-1 BCA (1977)
Samkal Mines, Inc.	DOTCAB 68-9A 71-1 BCA (1971)
Carson Linebaugh, Inc.	ASBCA 11384 67-2 BCA (1967)
Klefstad Engineering, Inc.	VACAB 522 66-1 BCA (1966)
Guy R. Allen	ASBCA 6896 1962 BCA
Coleman Electric Co.	ASBCA 4895 58-2 BCA (1958)
Allied Contractors Inc.	149 Ct. Cl. 671 277 F.2d 464 (1960)
American Dredging Co.	ENGBCA 2920 72-1 BCA (1972)

Hunkin Conkey Constr. Co. v. U.S.	198 Ct. Cl. 638 461 F.2d 1270 (1972)
United Contractors v. U.S.	177 Ct. Cl. 151 368 F.2d 585 (1966)
Morrison-Knudsen Co. v. U.S.	184 Ct. Cl. 661 397 F.2d 826 (1968)
Fehlhaber Corp. v. U.S.	138 Ct. Cl. 571 151 F. Supp. 817
Jack Crawford Contr. Co.	GSBCA 4089 75-2 BCA (1975)
Horner Constr. Co.	ASBCA 5335 59-2 BCA (1959)
William A. Smith Contracting Co. v. U.S.	188 Ct. Cl. 1062 412 F.2d 1325 (1969)
Jefferson Constr. Co. v. U.S.	176 Ct. Cl. 1363 364 F.2d 420 386 U.S. 914 (1966)
Schutt Constr. Co. v. U.S.	173 Ct. Cl. 836 353 F.2d 1018 (1965)
AFGO Engineering Corp.	VACAB 1236 79-2 BCA (1979)
Perini Corp. v. U.S.	180 Ct. Cl. 768 381 F.2d 403 (1967)
Quintana Constr. Co.	IBCA 1028-4-74 75-2 BCA (1975)
Gregg, Gibson & Gregg, Inc.	ENGBCA 3041 71-1 BCA (1971)
A. S. Horner, Inc.	AGBCA 76-145 79-1 BCA (1978)
Brezina Constr., Inc.	ENGBCA 3215 75-1 BCA (1974)
C. H. Leavall & Co.	ENGBCA 3492 75-2 BCA (1975)

John Murphy Constr. Co.	AGBCA 418 79-1 BCA (1979)
N. Fiorito Co. v. U.S.	189 Ct. Cl. 215 416 F.2d 1284 (1969)
Victory Constr. Co. v. U.S.	206 Ct. Cl. 274 510 F.2d 1379 (1975)
Victory Constr. Co.	ENGBCA 3009 69-2 BCA (1969)
Cf. Carvel Walker	ENGBCA 2744 78-1 BCA (1977)
Cottrell Engineering Corp.	ENGBCA 3038 70-2 BCA (1970)

APPENDIX E

DSCAS MANUAL

DSCAS USER'S MANUAL

E.1 Getting Started with DSCAS

The Differing Site Condition Analysis System (DSCAS) is a prototype expert system for the analysis of a differing site condition claim. The user's manual presented here is intended to help familiarize the user with the use of DSCAS. The presentation of the DSCAS given in this manual supplements the discussion contained in Chapter X by outlining the process used to begin and end a session and by presenting a sample session. This manual assumes that the user has a familiarity with both the UNIX operating system and the ROSIE programming environment, therefore, the how and why pertaining to operations within either of these environments will not be attempted.

The following discussion presents the steps necessary to get into the DSCAS environment. The host machine for DSCAS is currently a VAX 11/780 running the UNIX operating system. Once the user is logged in he must know the pathnames to both the LasII directory as well as the ROSIE-EXE file. Currently the alias "dscas" will place the user in the LasII directory which contains all the files necessary for DSCAS. Once the user is in the LasII directory he need only use the alias "rosie" to begin execution of the ROSIE.EXE file. ROSIE will begin by displaying a header, similar to the header shown in the sample session presented in the next session, followed by the ROSIE prompt <1>.

At this point the user is in the ROSIE environment, however. DSCAS has not been loaded yet. To load DSCAS the STARTUP file must be loaded using the "Load STARTUP." command (Note: A ROSIE command will not be executed unless the command or sentence ends with a "." or a "?"). The STARTUP file will be loaded at this time, displaying the rulesets which have been loaded and then asking the question "Would you like to run an analysis?" This question should be answered with a "Yes" or a "No." If the "yes" response is given all the necessary files for the analysis will be loaded and DSCAS will begin the analysis process. Otherwise the user will be politely returned to the ROSIE environment. To begin DSCAS directly from the ROSIE environment after the files have been loaded the user should use the following two commands: 1) "deactivate." which replaces the active database with the global database which contains the information telling DSCAS that all the necessary files are loaded and 2) "Go get-started." which starts the analysis process.

As DSCAS is being loaded the user will be kept aware of the sequence of events by messages displayed by DSCAS as well as the names of each ruleset as it is loaded. The loading process takes a few minutes . . . (Note: Each time the equivalent of a screenfull of information has been displayed the terminal will beep, hitting the space bar will continue the loading process). Once all files have been loaded DSCAS will introduce itself and ask if the user wishes to see additional instructions. The

results of answering "Yes" or "No" to this question should be intuitively clear.

Once DSCAS is ready to begin the analysis it will let the user know. At this point the user will be asked whether or not he wishes to continue a previous session. The "Yes" response will cause DSCAS to display a list of the viable previous sessions which can be loaded and request that the user select one. The method of selecting one of the choices is explained when the question is displayed. DSCAS then loads the indicated previous session. If "No" is the response DSCAS will request that the user assign a name to the current session to allow future reference to the session. (Note: the response to this question is not foolproof, seven or less characters should be used. Failure to abide by this may cause problems which will place the user back in the ROSIE environment. This is the only question with a problem of this type.)

After the user has either selected a previous session or assigned a name to the current session DSCAS asks the user whether or not he desires to keep a transcript of the session. If the user answers "Yes" a dribble file (see Fain et al. 1981) is created having the same name as the current session. The user should be alerted to a problem which may be encountered when a dribble file is being kept. If the "DELETE" key is used more than four or five times during the session the user will be dropped into the INTERLISP environment and given an error message. The 6 or ^

followed by a return will place the user back into DSCAS at the point which the "DELETE" key was pressed, i.e., DSCAS will be expecting a response to the question as if nothing happened. This problem results from a bug in INTERLISP, therefore, it is not recommended that new users keep transcripts of their sessions until they are familiar with DSCAS. From this point on the analysis is very user friendly and does not require any supporting explanation, therefore, the following sample session has been provided. The DSCAS prompt is a ":" followed by two spaces. At times answers entered will not be displayed immediately or DSCAS will seem to run very slowly, especially between asking questions. Be patient, DSCAS has a lot to think about.

E.2 Sample Session

The sample session presented here is the printout of the TEST1.ORIG file which is a transcript of the original TEST1 session, i.e., the session in which the questions were initially asked. Since the viable responses have been discussed in Chapter X and are explained in the on-line instructions they will not be reiterated here. The user can quite easily determine the effect of the various responses by comparing the answers with the corresponding assertion in the ANSWERS database displayed at the end of the session. Also, the effect of implementing special responses, such as the "-", the "?" and the "q", is exemplified by a number of questions.

TM
[Rosie Version 2.3 1-DEC-83 19:42:53]

The name of this session shall be: TEST1

You may perform any one of the following analyses:

DSC Assertion Final Payment Notice Form
Notice Promptness Responsible Receiver Government Prejudiced

Contract Obligation Excluded Conditions After-bid Conditions

Express-implied Conditions Contract Indications
Latent Deficiencies Standard Conditions
Site Inspection

Superior Knowledge-I Superior Knowledge-II
Reliance I Reliance II
Material Difference-I Material Difference-II
Exculpatory Langauge-I Exculpatory Language-II

Complete Analysis No Module

I will begin the analysis as soon as you have selected any ONE
of the options listed. Please enter the selected option
EXACTLY as it appears in the list followed by a '.'.
: Complete Analysis.

SESSION IN PROGRESS

[1] Has a differing site condition been asserted by one of
your contractors? (Yes/No)
: Yes

[2] On what date did the DSC-ASSERTION occur (month day, year)
: April 3, 1983

[3] Has final payment been made? (Yes/No)
: No

[4] What form of notice was given to the government by
the contractor? (Select one of the following.)
a. Written notice was given.
b. Oral notice was given.
c. Other events surrounded notice.

: b

[5] On what date did the ORAL-NOTICE occur (month day, year)
: -

[6] When was notice given in relation to the progression of work?
a. Notice given before contractor disturbed the condition.
b. Notice given before contractor performed remedial work.
c. Both a and b.
: b

[7] Is there evidence that the contractor should have been aware of the differing site condition sooner? (Yes/No)
: No

[8] To whom was notice of the differing site condition given?
a. The contracting officer
b. An authorized representative of the contracting officer
c. Any other employee of the government
d. None of the above
: c

[9] Was this other government employee cognisant of the implications of the condition? (Yes/No)
: Yes

[10] Did this other government employee communicate his awareness of the DSC to the CO? (Yes/No)
: Yes

[11] Is the government able to show that evidence against the claim was obscurred due to the passage of time? (Yes/No)
: No

[12] Is the government able to show that proper notice would have resulted in cheaper resolution of the condition by CO? (Yes/No)
: -

[13] Can the government show that it suffered any other prejudice through inadequacy of or lack of notice? (Yes/No)
: No

It is uncertain whether or not the government can
prove that it suffered prejudice from inadequacy of
or lack of notice.

[14] Who expressly assumed obligation for or risk of the condition in the contract?

a. The government
b. The contractor
c. Both the government and the contractor
d. Neither the government nor the contractor

: d

[15] Is the condition in question directly related to the physical conditions at the work site? (Yes/No)

: Yes

[16] Is the condition a static physical part of the work site? (Yes/No)

: -

[17] Did the condition occur before or after the contract award?

: ?

You may answer this question in the following manner:

Before -- If the condition occurred before the contract was awarded.

After -- If the condition occurred after the contract was awarded.

Also you may answer it with a '-' if the answer is unknown.

: Before

[18] Were there affirmatively expressed statements in the contract concerning the relevant subsurface or latent conditions at the site? (Yes/No)

: Yes

There were indications regarding the condition

in the contract, therefore, a Type I condition is
the most probable avenue of recovery.

[19] Were there latent deficiencies in the contract indications? (Yes/No)
: Yes

[20] Was the physical data made a part of the contract? (Yes/No)
: -

[21] Did the GOVERNMENT know about the CONTRACT-DEFICIENCY? (Yes/No)
: -

[22] Was the contractor in possession of unilateral superior or imputed knowledge that the actual conditions patently differed from the indications in the contract? (Yes/No)
: No

[23] Would simple inquiries by the contractor have revealed the condition to be contrary to the CONTRACT-INDICATIONS? (Yes/No)
: ?

Sorry no explanation for questions with yes/no answers.
: -

[24] Did any of the following acts of the government hinder the site inspection?

a. Access to site was denied
b. Inadequate time was allowed
c. No act of the government hindered inspection
: c

[25] Did the contractor conduct a site inspection? (Yes/No)
: Yes

[26] Was the inspection conducted by the contractor a reasonable inspection, i.e., one comparable to that expected from a reasonable, prudent contractor experienced in that particular field of work? (Yes/No)
: q

******************* SESSION INTERRUPTED *******************
(by user)

Do you wish to see the results from this session? (Yes/No)
: Yes

Which of the following results do you wish to see?

a. The answers given to the questions
b. The unknown information
c. The assumptions which have been made
d. The conclusions which have been reached
e. All of the above
f. None of the above

: e

These are the results of this session:

The answers:
[ANSWERS Database]
CONDITION did occur before CONTRACT AWARD.
CONTRACTOR did not have POTENTIAL-TO-KNOW CONDITION SOONER.
GOVERNMENT EMPLOYEE did understand IMPLICATIONS.
GOVERNMENT EMPLOYEE did communicate DSC AWARENESS TO-CO.
PASSAGE-OF-TIME did not obscure EVIDENCE.
NOTHING-ELSE did cause PREJUDICE.
GOVERNMENT did not expressly ASSUME OBLIGATION-RISK.
CONTRACTOR did not expressly ASSUME OBLIGATION-RISK.
CONTRACT did contain STATEMENTS CONCERNING CONDITION.
CONTRACT INDICATIONS did contain LATENT DEFICIENCIES.
GOVERNMENT did not hinder INSPECTION.
CONTRACTOR did conduct SITE-INSPECTION.
<APRIL 3, 1983> is a date-of-dsc-assertion.
<UNKNOWN> is a date-of-oral-notice.
NOTICE was received-by OTHER GOVERNMENT EMPLOYEE.
CONTRACTOR was not aware-of CONTRACT-DEFICIENCY.
DSC was asserted.
FINAL-PAYMENT was not made.
FORM-OF-NOTICE was oral.
NOTICE was given before REMEDIAL WORK PERFORMED.
CONDITION is directly-related-to-physical-conditions.

The unknowns:
[UNKNOWNS Database]
IF-IMPROPER-NOTICE did cause ADDITIONAL COST.
IF-CONTRACT INFORMATION did contain PHYSICAL DATA.
IF-SIMPLE INQUIRY did give POTENTIAL TO-KNOW CONTRARY CONDITIONS.
IF-GOVERNMENT was aware-of CONTRACT-DEFICIENCY.
IF-CONDITION is static PHYSICAL PART OF-WORK-SITE.

The assumptions:
[ASSUMPTIONS Database]
IMPROPER-NOTICE did not cause ADDITIONAL COST.
CONTRACT INFORMATION did not contain PHYSICAL DATA.
SIMPLE INQUIRY did not give POTENTIAL TO-KNOW CONTRARY CONDITIONS.
GOVERNMENT was not aware-of CONTRACT-DEFICIENCY.
CONDITION is static PHYSICAL PART OF-WORK-SITE.

The conclusions:
[REPORTS Database]
CONTRACTOR did constructively COMPLY with NOTICE-REQUIREMENT.
GOVERNMENT did receive NOTICE-OF-CONDITION.
CONTRACT did not assign OBLIGATION-RISK.
CONTRACT did contain INDICATIONS CONCERNING CONDITION.
PROMPTNESS_OF_NOTICE is an in_doubt.
NOTICE was given-to RESPONSIBLE RECEIVER.
NOTICE was prompt.
GOVERNMENT PREJUDICE is uncertain.

Which of the other results do you want to see?

a. The answers given to the questions
b. The unknown information
c. The assumptions which have been made
d. The conclusions which have been reached
e. All of the above
f. None of the above

: f

Do you wish to continue our analysis? (Yes/No)
Note: If you choose not to I will quit this session.
: Yes

******************* SESSION RESTARTING *******************

Since you interupted me I'll have to backtrack
a bit. Let's see as I remember....

The government did not hinder the site inspection
in any way.

The contractor did conduct a site inspection.

[26] Was the inspection couducted by the contractor a reasonable inspection, i.e., one comparable to that expected from a reasonable, prudent contractor experienced in that particular field of work? (Yes/No)
: Yes

[27] Would the actual condition have been discernable by a layman contractor performing a reasonable inspection? (Yes/No)
: No

[28] Did the contractor reasonably rely on DEFICIENT-INDICATIONS-IN-CONTRACT when preparing his bid causing him to suffer prejudice through this reliance? (Yes/No)
: Yes

[29] Was the difference between actual and INDICATED conditions a material difference? (Yes/No)
: Yes

Relief is probably available through the Type I DSC clause because of deficient indications in the contract.

[30] Is there an exculpatory clause denying any government liability and responsibility for actual conditions different from those indicated in the contract documents? (Yes/No)
: Yes

[31] Is the exculpatory clause specific to the DSC clause and to the condition encountered? (Yes/No)
: Yes

[32] Are the language and intent of the exculpatory clause clear and unambiguous? (Yes/No)
: -

I believe that the excupatory clause
is probably valid for this case.

Thinking.......

**** Entitlement will probably not be allowed under DSC claim. ****

I believe that entitlement will probably not be allowed because I concluded that the exculpatory-clause is probably valid.

The following rule proved to be true causing me to draw the above conclusion:

```
If
('exculpatory-clause is specific-to DSC clause'
                    is true in answers
       or 'exculpatory-clause is specific-to DSC clause'
                    is true in assumptions)
       and ('exculpatory-clause is clear-and-unambiguous'
                    is true in answers
         or 'exculpatory-clause is clear-and-unambiguous'
                    is true in assumptions)
      then the exculpatory-clause is probably valid.
```

The following statements contained by the rule are true:

[1] exculpatory-clause is specific-to DSC clause (in answers)

[2] exculpatory-clause is clear-and-unambiguous (in assumptions)

********************** END OF ANALYSIS **********************
(You should receive my bill in 7-10 days.)
(PROMPT PAYMENT IS APPRECIATED)

Do you wish to see the results from this session? (Yes/No)
: Yes

Which of the following results do you wish
to see?

a. The answers given to the questions
b. The unknown information

c. The assumptions which have been made
d. The conclusions which have been reached
e. All of the above
f. None of the above

: e

These are the results of this session:

The answers:
[ANSWERS Database]
CONDITION did occur before CONTRACT AWARD.
CONTRACTOR did not have POTENTIAL-TO-KNOW CONDITION SOONER.
REASONABLE INSPECTION did not have POTENTIAL-TO-REVEAL CONDITION.
GOVERNMENT EMPLOYEE did understand IMPLICATIONS.
GOVERNMENT EMPLOYEE did communicate DSC AWARENESS TO-CO.
PASSAGE-OF-TIME did not obscure EVIDENCE.
NOTHING-ELSE did cause PREJUDICE.
GOVERNMENT did not expressly ASSUME OBLIGATION-RISK.
CONTRACTOR did not expressly ASSUME OBLIGATION-RISK.
CONTRACT did contain STATEMENTS CONCERNING CONDITION.
CONTRACT INDICATIONS did contain LATENT DEFICIENCIES.
GOVERNMENT did not hinder INSPECTION.
CONTRACTOR did conduct SITE-INSPECTION.
CONTRACTOR did suffer PREJUDICE through RELIANCE.
CONTRACT does contain EXCULPATORY-CLAUSE.
<APRIL 3, 1983> is a date-of-dsc-assertion.
<UNKNOWN> is a date-of-oral-notice.
NOTICE was received-by OTHER GOVERNMENT EMPLOYEE.
CONTRACTOR was not aware-of CONTRACT-DEFICIENCY.
SITE-INSPECTION was reasonable INSPECTION.
EXCULPATORY-CLAUSE is specific-to DSC CLAUSE.
DSC was asserted.
FINAL-PAYMENT was not made.
FORM-OF-NOTICE was oral.
NOTICE was given before REMEDIAL WORK PERFORMED.
DIFFERENCE was material.
CONDITION is directly-related-to-physical-conditions.

The unknowns:
[UNKNOWNS Database]
IF-IMPROPER-NOTICE did cause ADDITIONAL COST.
IF-CONTRACT INFORMATION did contain PHYSICAL DATA.
IF-SIMPLE INQUIRY did give POTENTIAL TO-KNOW CONTRARY CONDITIONS.
IF-GOVERNMENT was aware-of CONTRACT-DEFICIENCY.
IF-CONDITION is static PHYSICAL PART OF-WORK-SITE.
IF-EXCULPATORY-CLAUSE is clear-and-unambiguous.

The assumptions:
[ASSUMPTIONS Database]
IMPROPER-NOTICE did not cause ADDITIONAL COST.
CONTRACT INFORMATION did not contain PHYSICAL DATA.

SIMPLE INQUIRY did not give POTENTIAL TO-KNOW CONTRARY CONDITIONS.
GOVERNMENT was not aware-of CONTRACT-DEFICIENCY.
CONDITION is static PHYSICAL PART OF-WORK-SITE.
EXCULPATORY-CLAUSE is clear-and-unambiguous.

The conclusions:
[REPORTS Database]
REASON_FOR NO ENTITLEMENT does exist.
CONTRACTOR did constructively COMPLY with NOTICE-REQUIREMENT.
GOVERNMENT did receive NOTICE-OF-CONDITION.
CONTRACT did not assign OBLIGATION-RISK.
CONTRACT did contain INDICATIONS CONCERNING CONDITION.
REASONABLE INSPECTION did not reveal CONDITION.
PROMPTNESS_OF_NOTICE is an in_doubt.
NOTICE was given-to RESPONSIBLE RECEIVER.
EXCULPATORY-CLAUSE is probably VALID.
NOTICE was prompt.
GOVERNMENT PREJUDICE is uncertain.
REASONABLE INSPECTION is not required-for-entitlement.

Which of the other results do you want to see?

a. The answers given to the questions
b. The unknown information
c. The assumptions which have been made
d. The conclusions which have been reached
e. All of the above
f. None of the above

: f

Do you wish to store this session? (Yes/No)
: Yes

Please enter your name or initials. (Please place a '-'
between initials and/or name.)
: TAK

Session has been saved as requested.

E.3 Wrapping Up the Analysis

Once a session is complete DSCAS will ask the user if he wishes to perform another analysis. If the response is "Yes" DSCAS will carry out the necessary preparations to begin a new session and the analysis cycle will begin again. If the response is "No" DSCAS will politely return the user to the ROSIE environment. At this point the user should logout of ROSIE with the "logout." command, this returns the user to the UNIX operating system. The DSCAS program is quite user friendly and by following the sequence of steps outlined by this manual the user will be insulated from unnecessary interaction with the ROSIE environment. However, to understand how DSCAS works and to gain the ability to modify and/or correct DSCAS's analysis process a working knowledge of the ROSIE environment is required. DSCAS is currently a competent analysis system which requires additional testing and modification. Therefore, continued experimentation and cautious development are encouraged to refine and enhance the current capabilities of DSCAS.

DISTRIBUTION

DTIC (12)

Kruppenbacher, Timothy A.
The application of artificial intelligence to contract management. --
Champaign, Ill : Construction Engineering Research Laboratory, 1984.
523 p. (Technical manuscript, P-166)
Thesis (M.S.) -- Univ. of Colorado.

1. Artificial intelligence. 2. Building--contracts and specifications.
I. Title. II. Series: Technical manuscript (Construction Engineering
Research Laboratory) ; P-166.

END

FILMED

11-84

DTIC